STATISTICAL INFERENCE FOR MACHINE LEARNING: FEATURE IMPORTANCE, UNCERTAINTY QUANTIFICATION AND INTERPRETATION STABILITY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Zhengze Zhou

May 2021

STATISTICAL INFERENCE FOR MACHINE LEARNING: FEATURE IMPORTANCE, UNCERTAINTY QUANTIFICATION AND INTERPRETATION STABILITY

Zhengze Zhou, Ph.D.

Cornell University 2021

Machine learning has become ubiquitous in many areas, including high-stake applications such as autonomous driving, financial forecasting and clinical decisions. However, many models are complex in nature and act as "black boxes", providing predictions but little insight as to how they were arrived at. In this thesis, we present our work from three different perspectives towards a better understanding of several types of machine learning models through the lens of statistical inference.

Tree-based methods, including decision trees, random forests and gradient boosting machines, are a popular class of nonparametric statistical model. They are widely used owing to their flexibility and superior performances. Many practitioners reply on some kind of feature importance measurements to examine model behavior. We propose a modification that corrects for split-improvement variable importance measures in random forests and other tree-based methods. These measurements have been shown to be biased towards increasing the importance of features with more potential splits. We show that by appropriately incorporating split-improvement as measured on out-of-sample data, this bias can be corrected yielding better summaries and screening tools.

Our next study focuses on understanding statistical properties and quantifying uncertainty for ensemble models. Tree-based ensembles like random

forests remain one such popular option for which several important theoretical advances have been made in recent years by drawing upon a connection between their natural subsampled structure and the classical theory of U-statistics. Unfortunately, the procedures for estimating predictive variance resulting from these studies are plagued by severe bias and extreme computational overhead. Here, we argue that the root of these problems lies in the structure of the resamples themselves. We develop a general framework for analyzing the asymptotic behavior of V-statistics, demonstrating asymptotic normality under precise regularity conditions and establishing previously unreported connections to U-statistics. Importantly, these findings allow us to produce a natural and efficient means of estimating the variance of a conditional expectation, a problem of wide interest across multiple scientific domains that also lies at the heart of uncertainty quantification for supervised learning ensembles. As an application, we apply this result to design a stopping rule for determining the ideal tree depth in model distillation.

Lastly, we investigate the stability for model explanation. Post hoc explanations based on perturbations are widely used approaches to interpret a machine learning model after it has been built. This class of methods has been shown to exhibit large instability, posing serious challenges to the effectiveness of the method itself and harming user trust. We propose a new algorithm called S-LIME, which utilizes a hypothesis testing framework based on central limit theorem for determining the number of perturbation points needed to guarantee stability of the resulting explanation. Experiments on both simulated and real world data sets are provided to demonstrate the effectiveness of our method.

BIOGRAPHICAL SKETCH

Zhengze Zhou was born in Zhoushan, an urbanized archipelago in eastern China, where he spent his childhood with his parents and grandmother. After one semester in the 7^{th} grade in 2007, he transferred to Zhenhai Jiaochuan Middle School in Ningbo and then attended Zhenhai High School for three years until 2012. He got admitted to School of Mathematical Sciences at Peking University thereafter.

He was immersed in the beauty and difficulty of math for two years before he declared a major in Statistics and Probability at the end of sophormore year. He was fortunate to receive May 4th Scholarship and Community Contribution Award during undergraduate study. He graduated in 2016 as an Outstanding Graduate, and his thesis work was supervised by Prof. Jinzhu Jia.

Zhengze joined the PhD program at Cornell University in Fall 2016 and moved to Ithaca, a gorgeous yet remote city on Cayuga Lake in New York. His general interests lies at the intersection of machine learning and statistical inference. His research work was advised by Prof. Giles Hooker. During five years at Cornell, he enjoyed the never-ending snow season while also developed professional skills through research, teaching and internship. He defended his thesis in April 2021.

Dedicated to my parents, Huizhong Zhou and Jiufei Shi.

ACKNOWLEDGEMENTS

This thesis would not be possible without the help and support from many people along the journal.

First and foremost, I would like to thank my advisor Giles Hooker. Our first connection dated back to February 2016 when he extended the PhD offer to me, and he was pretty helpful even before I came to Cornell. Since I started to work with him in the summer of 2017, he has been constantly available for advice and encouragement. I was impressed by his broad knowledge and deep insights as a statistician. To me, Giles has been and will always be a great mentor and friend.

I would also like to thank my committee members Madeleine Udell and Kilian Weinberger. Madeleine's group meeting has been a wonderful venue for me to gain new knowledge and seek valuable feedback. I also enjoyed a lot while taking Kilian's course on machine learning, and his high standard for presentation quality has inevitably made me better.

Furthermore, I want to express my gratitude to collaborators: Yichen Zhou, Lucas Mentch and Fei Wang; to my industry mentors: Anup Kotalwar, Kai Yuan and Dingxiong Deng; to an awesome statistics PhD cohort, and other faculty members and personnel in the department.

Finally, I am grateful for the support from my parents, family members and all my friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 A Primer on Machine Learning and Statistical Inference

Machine learning has made its appearance in everyday life: the spam filter in your email box, the search results returned based on your Google query, a virtual assistant like Siri or Alexa, just to name a few. In essense, machine learning is the practice of using algorithms to learn through experiences. Rather than explicitly tell a program what to, you teach the computer to develop an algorithm based on previous experience (i.e., data) to improve itself and complete the task.

There are three major types of machine learning:

- **Supervised learning** is to learn a function which maps from input to output from labeled data. The two main types of supervised learning are classification and regression. Spam filter is an example of the classification task, where an email needs to be classified as spam or not-spam. Regression involves continuous outputs such as predicting housing prices based on a few attributes.
- **Unsupervised learning** is a type of algorithm which learns patterns from unlabeled data. Two of the main methods are clustering and principal component analysis. The goal of clustering algorithms is to group or segment datasets based on similar properties. Principal component analysis is the process of computing the principal components and thus is usually used as a means for dimensional reduction.

- **Reinforcement learning** is concerned with how an agent interacts with the environment, collects feedback, and improves its own decision making process. Unlike supervised and unsupervised learning, reinforcement learning does not aim to find a "correct" output for a given instance. Instead the focus is to improve how the agent interacts with environment such that the rewards it receives in the long run can be maximized.

In this thesis we focus on supervised learning. Mathematically, suppose one has a set of n training examples denoted by $\mathcal{D} = \{z_1 = (\boldsymbol{x}_1, y_1), z_2 = (\boldsymbol{x}_2, y_2), \ldots, z_n = (\boldsymbol{x}_n, y_n)\}$ where $\boldsymbol{x}_i$ is the feature vector of i-th sample and y_i is the corresponding label, a supervised learning algorithm seeks a function $g : X \to Y$ which maps from the input space X to the output space Y. The function g is chosen from some space of possible functions denoted by $\mathcal{G}$, usually called hypothesis space.

For example in the spam filtering case, $\boldsymbol{x}_i$ denotes some features related to an email such as bag-of-words, and $y_i \in \{0, 1\}$ denoting whether this email is spam or not. To predict housing prices, $\boldsymbol{x}_i$ is a vector collecting information like square feet and number of bedrooms, $y_i \in \mathcal{R}$ being the price of the house. If we choose $\mathcal{G}$ to be the space of linear functions, then we'll learn a linear relationship between $\boldsymbol{x}_i$ and y; on the other hand, if we use more complex models such as random forests, then the hypothesis space $\mathcal{G}$ contains any function representable by an ensemble of decision trees. We'll provide more details in each chapter when it deals with certain model classes.

Now let's briefly introduce the concept of statistical inference. We'll build on the notations we just introduced. Assume all data in $\mathcal{D}$ comes from some *unknown* distribution $\mathcal{F}_Z$: $z_1, z_2, \ldots, z_n \overset{i.i.d.}{\sim} \mathcal{F}_Z$. A typical question we might ask

is:

Given a sample $z_1, z_2, \ldots, z_n$ drawn from an unknown distribution $\mathcal{F}_Z$, how to estimate $\mathcal{F}_Z$, or some quantities related to $\mathcal{F}_Z$?

In a more general sense, z_i can represent a pair of values $(\boldsymbol{x}_i, y_i)$ as in the supervised learning case, or just a scalar value.

There are three types of common statistical inference problems.

- **Point estimation** aims to find a particular quantity of interest. We can write the estimate as $\hat{\theta}_n = g(z_1, z_2, \ldots, z_n)$ for some function g. Here the true unknown quantity θ can be population mean, or parameters related to the underlying distribution $\mathcal{F}_Z$.
- **Confidence interval**. Instead of producing a point estimate, one may need a range of plausible values which contains the actual value with high probability. Given a confidence level α, a $1-\alpha$ confidence interval for θ is an interval (l_n, u_n) such that $P(\theta \in (l_n, u_n)) \geq 1-\alpha$.
- **Hypothesis testing** is the process of distinguishing between the null hypothesis H_0 and the alternative hypothesis H_1. Practically, a test statistic T is calculated based on the samples $z_1, z_2, \ldots, z_n$ and if T exceeds some threshold we will conclude to reject the null hypothesis. Statistical hypothesis testing is widely used in survey or experiment studies.

For a supervised learning problem, usually we can view it as a point estimation task. If the hypothesis space $\mathcal{G}$ can be parameterized by θ (such as linear models or neural networks), fitting a model is equivalent to finding a point estimate $\hat{\theta}$. For nonparametric case (such as tree ensembles), the model prediction

of a given instance x is the point of interest, where the true unknown quantity is $\theta = E_F[y|x]$.

At this point, it is worth mentioning one important philosophical difference between machine learning and statistical inference. Machine learning tries to make the most accurate predictions. Statistical inference is more about finding the relationship between variables, which also takes predictions into account by studying the statistical properties therein.

1.2 Roadmap

The topics in this thesis lie at the intersection of machine learning and statistical inference. Starting from some practical machine learning problems, we try to understand its behavior from the perspective of statistical inference. This section provides a roadmap where we introduce the motivation behind our work and present a high level summary of each chapter that follows.

Tree-based methods, such as random forests or gradient boosting decision trees, are arguably the most popular machine learning algorithms deployed in industry. However, large ensembles of trees act as "black boxes", providing predictions but little insight as to how they were arrived at. One of the approaches taken by practitioners is to look at variable importance scores. Previous studies have shown that split-improvement variable importance measures are biased towards increasing the importance of features with more potential splits. In Chapter 2, we show that by appropriately incorporating split-improvement as measured on out-of-sample data, this bias can be corrected yielding better summaries and screening tools. We also prove the proposed method is statistically

unbiased for features uncorrelated with the response.

In Chapter 3, we shift our attention to the predictions generated by ensembles, where the quantity of interest is $\theta = E_F[y|x]$. Machine learning aims to build algorithms which can get better estimate $\hat{\theta}$ of θ, while we focus on asymptotic properties of $\hat{\theta}$. Several important theoretical advances have been made in recent years by drawing upon a connection between the subsampled structure of ensembles and the classical theory of U-statistics. But the theory is unsatisfactory from two perspectives: contrary to common practice, it requires sampling without replacement; the procedures for estimating predictive variance resulting from these studies are plagued by severe bias and extreme computational overhead. In this work, we develop a general framework for analyzing the asymptotic behavior of V-statistics, demonstrating asymptotic normality under precise regularity conditions and establishing previously unreported connections to U-statistics. In addition, we propose an efficient variance estimation algorithm which also synthesizes most of the existing methods. As an application, we apply this result to design a stopping rule for determining the appropriate tree depth in model distillation.

Chapter 4 focuses on the stability of model explanation. Post hoc explanations based on perturbations are widely used approaches to interpret a machine learning model after it has been built. This class of method usually involves generating synthetic samples, and as a result exhibits large instability due to the randomness introduced. An unstable interpretation method can hardly be trusted by practitioners. We propose S-LIME, which utilizes a hypothesis testing framework based on central limit theorem for determining the number of perturbation points needed to guarantee stability of the resulting explanation.

We demonstrate its effectiveness by applying S-LIME on several real world data sets to explain different model types.

At the core of this thesis is a central theme: to better understand machine learning with the tool of statistical inference: we develop new feature importance measures and prove its statistical unbiasedness (Chapter 2); we analyze the asymptotics of ensemble predictions and propose efficient algorithms for variance estimation (Chapter 3); using hypothesis testing, we develop a stabilized post hoc model explanation method (Chapter 4). In the final chapter, we discuss some possible future directions.

CHAPTER 2

UNBIASED MEASUREMENT OF FEATURE IMPORTANCE IN TREE-BASED METHODS

2.1 Introduction

This chapter examines split-improvement feature importance scores for tree-based methods. Starting with Classification and Regression Trees (CART; Breiman et al., 1984) and C4.5 (Quinlan, 2014), decision trees have been a workhorse of general machine learning, particularly within ensemble methods such as Random Forests (RF; Breiman, 2001) and Gradient Boosting Trees (Friedman, 2001). They enjoy the benefits of computational speed, few tuning parameters and natural ways of handling missing values. Recent statistical theory for ensemble methods (e.g. Denil et al., 2014; Scornet et al., 2015; Mentch and Hooker, 2016; Wager and Athey, 2018; Zhou and Hooker, 2018) has provided theoretical guarantees and allowed formal statistical inference. Variants of these models have also been proposed such as Bernoulli Random Forests (Yisen et al., 2016; Wang et al., 2017) and Random Survival Forests (Ishwaran et al., 2008). For all these reasons, tree-based methods have seen broad applications including in protein interaction models (Meyer et al., 2017), in product suggestions on Amazon (Sorokina and Cantú-Paz, 2016) and in financial risk management (Khaidem et al., 2016).

However, in common with other machine learning models, large ensembles of trees act as "black boxes", providing predictions but little insight as to how they were arrived at. There has thus been considerable interest in providing tools either to explain the broad patterns that are modeled by these methods,

or to provide justifications for particular predictions. This chapter examines variable or feature[1] importance scores that provide global summaries of how influential a particular input dimension is in the models' predictions. These have been among the earliest diagnostic tools for machine learning and have been put to practical use as screening tools, see for example Díaz-Uriarte and De Andres (2006) and Menze et al. (2009). Thus, it is crucial that these feature importance measures reliably produce well-understood summaries.

Feature importance scores for tree-based models can be broadly split into two categories. Permutation methods rely on measuring the change in value or accuracy when the values of one feature are replaced by uninformative noise, often generated by a permutation. These have the advantage of being applicable to any function, but have been critiqued by Hooker (2007); Strobl et al. (2008); Hooker and Mentch (2019) for forcing the model to extrapolate. By contrast, in this chapter we study the alternative split-improvement scores (also known as Gini importance, or mean decrease impurity) that are specific to tree-based methods. These naturally aggregate the improvement associated with each node split and can be readily recorded within the tree building process (Breiman et al., 1984; Friedman, 2001). In Python, split-improvement is the default implementation for almost every tree-based model, including **RandomForestClassifier**, **RandomForestRegressor**, **GradientBoostingClassifier** and **GradientBoostingRegressor** from **scikit-learn** (Pedregosa et al., 2011a).

Despite their common use, split-improvement measures are biased towards features that exhibit more potential splits and in particular towards continuous features or features with large numbers of categories. This weakness was

[1]We use "feature", "variable" and "covariate" interchangeably here to indicate individual measurements that act as inputs to a machine learning model from which a prediction is made.

already noticed in Breiman et al. (1984) and Strobl et al. (2007) conducted thorough experiments followed by more discussions in Boulesteix et al. (2011) and Nicodemus (2011)[2]. While this may not be concerning when all covariates are similarly configured, in practice it is common to have a combination of categorical and continuous variables in which emphasizing more complex features may mislead any subsequent analysis. For example, gender will be a very important binary predictor in applications related to medical treatment; whether the user is a paid subscriber is also central to some tasks such as in Amazon and Netflix. But each of these may be rated as less relevant to age which is a more complex feature in either case. In the task of ranking single nucleotide polymorphisms with respect to their ability to predict a target phenotype, researchers may overlook rare variants as common ones are systematically favoured by the split-improvement measurement. (Boulesteix et al., 2011).

We offer an intuitive rationale for this phenomenon and design a simple fix to solve the bias problem. The observed bias is similar to overfitting in training machine learning models, where we should not build the model and evaluate relevant performance using the same set of data. To fix this, split-improvement calculated from a separate test set is taken into consideration. We further demonstrate that this new measurement is unbiased in the sense that features with no predictive power for the target variable will receive an importance score of zero in expectation. These measures can be very readily implemented in tree-based software packages. We believe the proposed measurement provides a more sensible means for evaluating feature importance in practice.

In the following, we introduce some background and notation for tree-based methods in Section 2.2. In Section 2.3, split-improvement is described in detail

[2]See `https://explained.ai/rf-importance/` for a popular demonstration of this.