UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

A Bachelor of Science thesis

Super Learners

and their oracle properties

Jinyang Liu

Supervised by Prof. Thomas Gerds
Co-supervised by Prof. Niels Richard Hansen
Department of Mathematical Sciences
University of Copenhagen, Denmark

Submitted: April 25, 2023

Contents

1 Introduction

In the context of regression, a natural goal is to obtain an regression function θ such that the L^2-risk or mean squared error $E(Y - \theta(X))^2$ for n observation $O = (Y, X)$ is minimal. It turns out that the conditional mean $x \mapsto E(Y \mid X = x)$ minimizes the squared error, but the task of identifying the conditional mean is challenging unless we know the underlying data-generating process $P \in \mathcal{P}$ for which $O \sim P$. We typically make certain assumptions about the statistical model, $\mathcal{P}$, in which we believe P resides. For instance, we might assume that $\mathcal{P}$ is a curved exponential family. In doing so we are able to identify through maximum likelihood techniques, the parameters of the distribution P and achieve an asymptotic convergence rate of $O(1/n)$.

However, under certain circumstances we might not be able to make such assumptions. For example, if we have limited data, identifying the model as an exponential family might not be appropriate. In such cases, it may be more suitable to utilize non-parametric and data-driven regression methods, such as tree-based algorithms like XGBoost or random forests, to estimate the conditional mean. However, the assumptions of these data-driven methods regarding $\mathcal{P}$ are not explicit, and they may not have probabilistic interpretations. We can nevertheless incorporate these methods as a part of our repertoire, but it is important that we can compare and choose the best method that most effectively accomplishes our goal.

The "super learner" is the answer to how we can effectively select the 'best' learner among the learners that we have in our library of learners. We will demonstrate that the cross-validation selector, which evaluates learners based on their cross-validated (empirical) risk and chooses the one with the lowest risk, is asymptotically equivalent to the oracle selector. The oracle selector identifies the learner with the lowest true risk – the theoretical risk achieved by knowing the true distribution.

The discrete super learner is then obtained by applying the cross-validation selector on our library of learners. The asymptotic result shows that the risk of the super learner will be the same as the learner selected by the oracle selector when the number of observations goes to ∞. The discrete super learner is not a fixed learner from our library, but rather depends on the available data. It represents the learner chosen by the cross-validation selector, which can vary depending on the amount of data at hand.

We first present the general theory and our goal, which is to estimate the conditional mean $E(Y \mid X = x)$ for a observation Y, X being a outcome-covariate pair. More specifically, we focus on the case where we regress on a binary outcome $Y \in \{0, 1\}$. The conditional expectation of Y given X exactly becomes the conditional probability $P(Y = 1 \mid X = x)$. The choice to focus on binary regression stems from its significance in various fields. For instance, in biomedicine, researchers might want to predict patient mortality upon administering a specific drug. The survival indicator for the patient is a binary outcome, and the regression $P(Y = 1 \mid X = x)$ could represent the probability of the patient's survival.

2 Background

Our setup closely models what is described in [VDL06] and [LD03]. Unless specified otherwise, our setup is as follows: Let statistical model $\mathcal{P}$ be given on the measurable space $(\mathcal{O}, \mathcal{A})$ where $\mathcal{O} = \{0, 1\} \times \mathcal{X}$ is our sample space for some $\mathcal{X} \subseteq \mathbb{R}^d$. We will consider the parameter set $\theta = \{\theta \mid \theta : \mathcal{X} \to [0, 1]\}$, which represents the set of regression functions that map from our covariates to the probability interval. We define the quadratic loss and

the corresponding risk that we wish to minimize

Definition 1 (Quadratic loss)**.** Let $\mathcal{O}$ be our sample space, and θ be the set of regression functions. Then the quadratic loss or L^2-loss, $L : \mathcal{O} \times \theta \to [0, \infty)$, for an observation $o \in \mathcal{O}$ and a regression function $\theta \in \Theta$ is defined as

$$L(o, \theta) = L((y, x), \theta) = (y - \theta(x))^2.$$

Our natural aim would be to find the optimal parameter value $\theta^* \in \Theta$ that minimizes the L^2-risk $R : \theta \to \mathbb{R}$ given as

$$R(\theta) := \int L(o, \theta) dP(o) = EL(O, \theta), \tag{1}$$

but this task is challenging as it requires us to know the data-generating process, P, which we do not have access to. It can be shown that the minimum risk is achieved by the conditional probability $x \mapsto P(Y = 1 \mid X = x)$.

Theorem 2. *Let $(\mathcal{O}, \mathcal{A}, P)$ be a probability space for some probability measure $P \in \mathcal{P}$. Let θ be the set of regression functions of the form $\theta : \mathcal{X} \to [0, 1]$. Let the loss function be the L^2-loss $L(o, \theta) = (y - \theta(x))^2$, then for the optimum θ^* defined as*

$$\theta^* := \arg\min_{\theta \in \Theta} R(\theta) = \arg\min_{\theta \in \Theta} \int L(o, \theta) dP(o),$$

it holds for an observation $O = (Y, X) \sim P$ that

$$\theta^*(x) = E(Y \mid X = x)$$

Proof. See [Gyö+02][ch. 1] □

It follows immediately that if Y is binary, then $E(Y \mid X = x) = P(Y = 1 \mid X = x)$. In the case where we do not have access to the data-generating process, we would instead want to provide an estimate $\hat{\theta}$ of θ^* as a function of the data that we have observed. Let $O_1 = (Y_1, X_1), \ldots, O_n = (Y_n, X_n) \in \mathcal{O}$ be i.i.d. observations distributed according to some $P \in \mathcal{P}$. Denote our data as $D_n = (O_1, \ldots, O_n)$. An estimate is the outcome of applying an estimator to the data

$$D_n \mapsto \hat{\theta}(D_n) \in \Theta,$$

The estimate is a regression function, that is

$$\mathcal{X} \ni x \mapsto \hat{\theta}(D_n)(x) \in [0, 1].$$

In the context of super learners we will refer the estimators as learners, in the sense that they learn from the data, the resulting regression function is then the fit of the learner to the data. We formalize these notions in the subsequent section.

Example 1 (Parametric statistical model). In the case where we have n-i.i.d. observations distributed according to some data-generating process $P \in \mathcal{P}$, we have that each $O_i = (Y_i, X_i)$, and X_i is a stochastic variable. The distribution P factorizes essentially into two parts, the conditional distribution of Y given X and the background distribution of X, so $P = P_{Y|X=x} \cdot P_X$. In this setup we are doing estimation with random design [Gyö+02].
We can formalize the setup as follows: $O = (Y, X) \sim P$, if Y is $\mathcal{B}(\mathbb{R})-\mathcal{B}(\mathbb{R})$ measurable and X is $\mathcal{F} - \mathcal{B}(\mathbb{R})$ measurable for some sigma-algebra $\mathcal{F}$ on $\mathcal{X}$, then a **generalized regression model** could be considered as parametrized family of distributions, $\mathcal{P} = \{P_\theta \mid \theta \in \Theta\}$, given that θ is finite-dimensional.
We can parametrize the conditional probability distributions for Y_1 given $X_1 = x$ as $\mathcal{Q} = \{Q_{\theta(x)} \mid \theta \in \Theta\}$ such that $Q_{\theta(x)}$ is a valid probability distribution on $\mathcal{B}(\mathbb{R})$ for each $x \in X$ and $\theta \in \Theta$. For a given $P_\theta \in \mathcal{P}$ there will exist a $Q_\theta \in \mathcal{Q}$ such that

$$P_\theta(Y \in A \mid X = x) = Q_{\theta(x)}(A) \qquad \text{for all } A \in \mathcal{B}(\mathbb{R}).$$

If we assume that X_1 is distributed according to some H_0 on $\mathcal{X}$, then the distribution P_θ over our observations (the joint over Y and X) will be

$$P_\theta(X \in A, Y \in B) = \int_A Q_{\theta(x)}(B) dH_0(x)$$

for every $A \in \mathcal{F}$ and $B \in \mathcal{B}(\mathbb{R})$.

3 The Discrete Super Learner

In the following section we introduce the terminology **learning algorithm** and **learners** in the context of learning from our data.

3.1 Learning algorithm

Definition 3 (Learning algorithm θ)**.** An learning algorithm is a measurable map $\theta : \mathcal{O}^n \to \theta$ for $n \in \mathbb{N}$.

We use the notation θ for the learning algorithm, which coincides with the notation for a regression function that resides in θ. Indeed, it makes sense in our context since for $D_n = (O_1, \ldots, O_n)$ being our data, we would like to express the outcome of applying a learning algorithm to our data, $\theta(D_n)$, we refer to that as the **fitted learner** which is in θ. We will furthermore assume that the learning algorithm is well defined for each $n \in \mathbb{N}$, and that permuting the observations have no effect on the outcome, i.e., the algorithm is symmetric in the observations.
However, we must remark that formally, $\theta(D_n)$ is a stochastic variable since D_n is stochastic. It is, therefore, a map from a background space, Ω, to the parameter space, θ. In practice, we would have observed $O_3(\omega) = o_1, \ldots, O_n(\omega) = o_n$ for a specific omega, and subsequently, we can fit our learning algorithm on $D_n(\omega) = (o_1, \ldots, o_n)$, which is a particular instance of a dataset. The fitted learner, $\theta(D_n(\omega))$, is a regression function in θ. The abuse in notation is analogous to stating that "$X \in \mathbb{R}$" for a real random variable X, even though this is not technically correct since X is a measurable map rather than a real number.

Example 2 (Parametric and nonparametric learning algorithms). An example of a parametric learner is logistic regression. In logistic regression we assume that conditional probability can be expressed as $x \mapsto P(Y = 1 \mid X = x) = \text{expit}(\beta x)$ for some $\beta \in \mathbb{R}^d$. The parameter β can be estimated via maximum likelihood.
Nonparametric learning algorithms such as gradient boosting, for example XGBoost, can also be used to estimate the regression function. These methods have a number of hyperparameters that can be tuned. These could include, number of boosted trees, depth of each tree, learning rates, etc. Gradient boosting seeks to iteratively optimize the algorithm approximating the data $x \mapsto f_m(x)$ at each step m by adding a new tree $h_m(x)$ such that $f_{m+1}(x) = f_m(x) + h_m(x)$ such that the mean squared error loss of the new learner, f_{m+1}, is less than the previous, f_m. [CG16]
Gradient boosting algorithms internally evaluate the loss of the fitted learner at each iterative step, seeking to minimize the loss by adding new trees that compensate for the loss of the previous trees. The fitted parameters of the resulting tree do not have probabilistic interpretations. Regardless, algorithms such as XGBoost have demonstrated that they are indeed capable of fitting on very complex datasets.

There is a one-to-one correspondence between our data $D_n = (O_1, \ldots, O_n)$ and the empirical measures over n observations on $(\mathcal{O}, \mathcal{A})$ defined as

$$P_n(A) = \frac{1}{n} \sum_{i=1}^{n} \delta_{O_i}(A) \qquad \text{for } A \in \mathcal{A}.$$

We can, therefore, write $\theta(P_n)$ as an alternative representation of the learner $\theta(D_n)$, by adjusting the notation slightly without introducing ambiguity. The motivation for using this notation will become clearer in the subsequent section, where we introduce the cross-validation selector.

3.2 Library of learners

We would now like to consider the scenario where we have a library (set) of learning algorithms, $\theta_1, \ldots, \theta_n$. From these algorithms, we can define the library of learners $\hat{\Theta} = \{\hat{\theta}_q = \theta_q(P_n) \mid 1 \leq q \leq k\}$. Once again, our natural goal is to find $\hat{\theta}$, that achieves the lowest risk among our learners, that is to find q such that $R(\hat{\theta}_q) = \min_{\hat{\theta} \in \hat{\Theta}} R(\hat{\theta})$. But as we have remarked before, this is not possible unless we know the data-generating process P, instead we can only provide an estimate $\hat{q}$ of q that is based on our data.

3.3 Cross-validation methodology

To provide the estimate $\hat{q}$ we have proceed via cross validation. In cross validation, we randomly split our data into a training set and a test set. Let the random binary vector $S = (S_1, \ldots, S_n) \in \{0, 1\}^n$ be independent of $O_1, \ldots, O_n$ such that $S_i = 0$ indicates that O_i should be in the training set and $S_i = 1$ indicates that O_i belongs to the test set. We can define the empirical distributions over these two subsets, $P^0_{n,S}$ and $P^1_{n,S}$ as

$$P^0_{n,S} = \frac{1}{n_0} \sum_{i:S_i=0} \delta_{O_i}$$

$$P^1_{n,S} = \frac{1}{n_1} \sum_{i:S_i=1} \delta_{O_i}$$

Where n_0, n_1 would be the number of S_i's that are marked 0 and 1 respectively.

Example 3 (Random splits). For $n = 9$ observations one could for example define the distribution of the random vector S as

$$P(S = (0,0,0,0,0,0,1,1,1)) = \frac{1}{3},$$
$$P(S = (0,0,0,1,1,1,0,0,0)) = \frac{1}{3},$$
$$P(S = (1,1,1,0,0,0,0,0,0)) = \frac{1}{3},$$

i.e. 3-fold cross-validation. In general for n observations we have 2^n ways of choosing which observations should be in the training set and in the validation set. It might not be desirable to define the discrete probabilities for S over $\{0,1\}^n$ simply as $\frac{1}{2^n}$ for each possible combination of training/validation data, since that would also include the combination where $n_1 = 0$. To ensure that we always have a certain amount of observations in our validation set, let $n_1 > 0$ be given, we see that there are $\binom{n}{n_1}$ ways of choosing both the validation and training set. We can therefore define the distribution of S as

$$P\left(S = s\right) = \binom{n}{n_1}^{-1} \qquad \text{for each } s \in \{0,1\}^n \text{ where } \sum_i s_i = n_1,$$

this procedure is also known as Monte Carlo cross-validation.

3.4 Risks and selectors

We now provide the formal definitions for the risks associated with our learners. Recall that the L^2-risk (1) was the integral of the loss with respect to data-generating process P. Upon observing our data D_n, we can define the empirical risk as the integral of the loss function with respect to P_n, as follows

$$\hat{R}(\theta) := \int L(o,\theta) dP_n(o).$$

Now let some split variable S be given for our data D_n, we are also interested in how our learner performs on the cross-validation data. Denote the risk of our learner on the cross-validation data as

$$\hat{R}_S(\theta) := \int L(o,\theta) dP^1_{n,S}(o).$$

The following definitions are analogous to what is stated in section 1 and 2 of [LD03]

Definition 4 (Conditional risk averaged over splits [VDL06])**.** Given the data D_n and some split-variable S. The averaged conditional risk of each learner in a specified library, $\hat{\Theta} = \{\theta_q(P^0_{n,S}) \mid 1 \leq q \leq k\}, k \in \mathbb{N}$, applied to our training data $P^0_{n,S}$ is

$$q \mapsto E_S R(\theta_q(P^0_{n,S})).$$

The expectation E_S is a simple average since S is discrete. Therefore, for a given q we have

$$E_S R(\theta_q(P^0_{n,S})) = \sum_{s \in \text{supp}(S)} R(\theta_q(P^0_{n,S=s})) \cdot P(S = s)$$

Definition 5 (Oracle selector)**.** Given the data D_n and some split variable S. The oracle selector depends on our data and is the index of the learner with the lowest averaged conditional risk given our data

$$\tilde{q} := \underset{1\leq q\leq k}{\arg\min}\, E_S R(\theta_q(P^0_{n,S}).$$

As we are unable to calculate the true conditional risk, we must proceed via cross-validation. Cross-validation replaces R with $\hat{R}_S$.

Definition 6 (Cross-validation risk)**.** Given the data D_n and some split-variable S. The cross-validation risk of a learner in a specified library, $\hat{\Theta} = \{\theta_q(P^0_{n,S}) \mid 1 \leq q \leq k\}, k \in \mathbb{N}$, applied to our training data $P^0_{n,S}$ is

$$q \mapsto E_S \hat{R}_S(\theta_q(P^0_{n,S})).$$

Definition 7 (Cross-validation selector [LD03])**.** Given the data D_n and some split variable S. The cross validation selector depends on our data and is the index of the learner with the lowest cross-validation risk given our data

$$\hat{q} := \underset{1\leq q\leq k}{\arg\min}\, E_S \hat{R}_S(\theta_q(P^0_{n,S})).$$

We are now ready to give the definition of the discrete super learner

Definition 8 (Discrete super learner)**.** Let $(O, \mathcal{A}, P)$ be a measure space for some $P \in \mathcal{P}$ where $\mathcal{P}$ is our statistical model. Let $D_n = (O_1, \ldots, O_n)$ be our data for each $O_i = (Y_i, X_i) \sim P$. For a split-variable S that specifies the cross-validation procedure, the **discrete super learner** created from some library of learners $\hat{\Theta} = \{\theta_q(P^0_{n,S}) \mid 1 \leq q \leq k\}, k \in \mathbb{N}$ is the learner chosen by the cross-validation selector fitted on the entire dataset

$$\mathcal{X} \ni x \mapsto \theta_{\hat{q}}(P_n)(x).$$

Remark: An alternative way of defining the super learner is $x \mapsto E_S \theta_{\hat{q}}(P^0_{n,S})(x)$, it might be that this definition is better since $\theta_{\hat{q}}(P_n)$ is technically a new learner in our library. It may, therefore, not be possible to fit the learning algorithm $\theta_{\hat{q}}$ on D_n due to the difference in the number of observations.

3.5 Oracle inequalities

We introduce the notation Pf for the integral $\int f dP$ of an integrable function f with respect to P. Additionally, if P_n represents the empirical measure of $O_1, \ldots, O_n$, we denote the empirical process indexed over an appropriate class of functions $\mathcal{F}$ as $G_n f = \sqrt{n}(P_n f - Pf)$. Furthermore, we extend this notation to $G^i_{n,S} f = \sqrt{n}(P^i_{n,S} - Pf)$ for the empirical processes that correspond to applying the empirical measure over either the training sample or validation sample.
In the following results we assume that a proper loss function $L : \mathcal{O} \times \Theta \to \mathbb{R}$ has been given.

Lemma 9 (Lemma 2.1 in [VDL06]). *Let G_n be the empirical process of an i.i.d. sample of size n from the distribution P. For $\delta > 0$ it holds that*

$$\begin{aligned} E_S \int L(o, \theta_{\hat{q}}(P^0_{n,S}))dP(o) \leq (1+2\delta)E_S \int L(o, \theta_{\tilde{q}}(P^0_{n,S}))dP(o) \\ + \frac{1}{\sqrt{n_1}} E_S \max_{1\leq q\leq k} \int L(o, \theta_q(P^0_{n,S}))d((1+\delta)G^1_{n,S} - \delta\sqrt{n_1}P)(o) \\ + \frac{1}{\sqrt{n_1}} E_S \max_{1\leq q\leq k} \int -L(o, \theta_q(P^0_{n,S}))d((1+\delta)G^1_{n,S} + \delta\sqrt{n_1}P)(o) \end{aligned}$$

Proof. See appendix □

Lemma 10 (Lemma 2.2 in [VDL06]). *Let G_n be the empirical process of an i.i.d. sample of size n from the distribution P and assume that $Pf \geq 0$ for every $f \in \mathcal{F}$ in some set of measurable functions $\mathcal{F}$. Then, for any Bernstein pairs $(M(f), v(f))$ and for any $\delta > 0$ and $1 \leq p \leq 2$,*

$$E \max_{f\in\mathcal{F}} (G_n - \delta\sqrt{n}P)f \leq \frac{8}{n^{1/p-1/2}} \log(1+\#\mathcal{F}) \max_{f\in\mathcal{F}} \left[\frac{M(f)}{n^{1-1/p}} + \left(\frac{v(f)}{(\delta Pf)^{2-p}} \right)^{1/p} \right].$$

The same upper bound is valid for $E \max_{f\in\mathcal{F}} (G_n + \delta\sqrt{n}P)(-f)$

Theorem 11 (Finite Sample Result: Theorem 2.3 in [VDL06]). *For $\theta \in \Theta$ let $(M(\theta), v(\theta))$ be a Bernstein pair for the function $o \mapsto L(o, \theta)$ and assume that $R(\theta) = \int L(o, \theta)dP(o) \geq 0$ for every $\theta \in \Theta$. Then for $\delta > 0$ and $1 \leq p \leq 2$ it holds that*

$$\begin{aligned} ER(\theta_{\hat{q}}(P^0_{n,S})) \leq & (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + \\ & (1+\delta)E\left(\frac{16}{n_1^{1/p}} \log(1+k) \sup_{\theta\in\Theta} \left[\frac{M(\theta)}{n_1^{1-1/p}} + \left(\frac{v(\theta)}{R(\theta)^{2-p}} \right)^{1/p} \left(\frac{1+\delta}{\delta} \right)^{2/p-1} \right] \right), \end{aligned}$$

where k is the number of learners in our library $\{\theta_q(P^0_{n,S}) \mid 1 \leq q \leq k\}$.

Proof. We will apply lemma 10 to the second and third terms on the left hand side of the inequality in lemma 9. Let $\mathcal{F} = \{o \mapsto L(o, \theta_q(P^0_{n,S})) \mid 1 \leq q \leq k\}$, be the collection of functions obtained by applying the loss L to each learner in our library of k learners. Note that $\mathcal{F} \subseteq \{o \mapsto L(o, \theta) \mid \theta \in \Theta\}$, and since $R(\theta) \geq 0$ for every $\theta \in \Theta$ it follows that $Pf \geq 0$ for every $f \in \mathcal{F}$.
For the second term we have

$$\begin{aligned} & \frac{1}{\sqrt{n_1}} E_S \max_{1\leq q\leq k} \int L(o, \theta_q(P^0_{n,S}))d((1+\delta)G^1_{n,S} - \delta\sqrt{n_1}P)(o) \\ & = \frac{1+\delta}{\sqrt{n_1}} E_S \max_{1\leq q\leq k} \int L(o, \theta_q(P^0_{n,S}))d(G^1_{n,S} - \frac{\delta}{1+\delta}\sqrt{n_1}P)(o), \end{aligned}$$

applying lemma 10 to the expression above yields

$$\begin{aligned}
&\frac{1+\delta}{\sqrt{n_1}} E_S \max_{1\le q\le k} \int L(o,\theta_q(P^0_{n,S}))d(G^1_{n,S} - \frac{\delta}{1+\delta}\sqrt{n_1}P)(o) \\
&\le \frac{1+\delta}{\sqrt{n_1}} \left(\frac{8}{n_1^{1/p-1/2}} \log(1+k) \max_{1\le q\le k} \left[\frac{M(\theta_q(P^0_{n,S}))}{n_1^{1-1/p}} + \left(\frac{v(\theta_q(P^0_{n,S}))}{(\frac{\delta}{1+\delta})^{2-p} R(\theta_q(P^0_{n,S}))^{2-p}} \right)^{1/p} \right] \right) \\
&\le \frac{1+\delta}{\sqrt{n_1}} \left(\frac{8}{n_1^{1/p-1/2}} \log(1+k) \sup_{\theta\in\Theta} \left[\frac{M(\theta)}{n_1^{1-1/p}} + \left(\frac{v(\theta)}{(\frac{\delta}{1+\delta})^{2-p} R(\theta)^{2-p}} \right)^{1/p} \right] \right) \\
&= (1+\delta)\frac{8}{n_1^{1/p}} \log(1+k) \sup_{\theta\in\Theta} \left[\frac{M(\theta)}{n_1^{1-1/p}} + \left(\frac{v(\theta)}{R(\theta)^{2-p}} \right)^{1/p} \left(\frac{1+\delta}{\delta} \right)^{2/p-1} \right]
\end{aligned}$$

Where for the third inequality we take the sup over Θ. We can also bound the third term with the same expression above. It is now immediate from lemma 9 that

$$\begin{aligned}
E_S \int L(o,\theta_{\hat{q}}(P^0_{n,S}))dP(o) \le (1+2\delta) E_S \int L(o,\theta_{\tilde{q}}(P^0_{n,S}))dP(o) \\
+ 2\cdot(1+\delta)\frac{8}{n_1^{1/p}} \log(1+k) \sup_{\theta\in\Theta} \left[\frac{M(\theta)}{n_1^{1-1/p}} + \left(\frac{v(\theta)}{R(\theta)^{2-p}} \right)^{1/p} \left(\frac{1+\delta}{\delta} \right)^{2/p-1} \right],
\end{aligned}$$

which was exactly what we set out to prove. □

3.6 Example: Binary Regression

Consider the case where we have i.id. observations $O_1 = (Y_1, X_1), \ldots, O_n = (Y_n, X_n)$ such that $Y_i \in \{0,1\}$ and $X \in \mathbb{R}^d$ distributed according some $P \in \mathcal{P}$. We would like to estimate the conditional expectation $\theta_0(x) = E(Y \mid X = x) = P(Y = 1 \mid X = x)$. Let $\theta = \{\theta \mid \theta : \mathcal{X} \to [0,1] \text{ measurable}\}$ and choose the quadratic loss function $L((Y,X),\theta) = (Y - \theta(X))^2$.

We observe that the quadratic loss is bounded by 1 for all choices of $\theta \in \Theta$ and $O \in \mathcal{O}$. It is stated in [VDL06, p. 7] that $M(\theta) = 1$ and $v(\theta) = \frac{3}{2}\int L(o,\theta)^2 dP(o)$ is a valid Bernstein pair for the function $o \mapsto L(o,\theta)$. It is also clear that $R(\theta) = \int L(o,\theta)dP(o) \ge 0$ since the loss function is positive. If we plug these numbers in theorem 11, then by using $p = 1$ and

$$ER(\theta_{\hat{q}}(P^0_{n,S})) \le (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + (1+\delta)E\left(\frac{16}{n_1}\log(1+k) \sup_{\theta\in\Theta} \left[M(\theta) + \frac{v(\theta)}{R(\theta)}\frac{1+\delta}{\delta} \right] \right)$$

In the equation provided, we observe that we can manipulate the following variables: sample size n, validation set size n_1, parameter $\delta > 0$, and the number of learners k. Assuming k remains constant, the validation set size n_1 could be either stochastic, depending on the split variable S, or fixed as a constant, as illustrated in example 3. For instance, we can set $n_1 = n/2$. By establishing a fixed value for n_1, we can drop the expectation in the second term.

The supremum on the left side of the equation might increase significantly because $R(\theta)$ could be very small. However, by carefully selecting the value of $v(\theta)$, we can avoid the fraction from growing too large. Note that for any $\theta \in \Theta$:

$$\frac{v(\theta)}{R(\theta)} = \frac{3}{2}\frac{\int L(o,\theta)^2 dP(o)}{\int L(o,\theta)dP(o)} \le \frac{3}{2}\frac{\int L(o,\theta)\cdot 1 dP(o)}{\int L(o,\theta)dP(o)} = \frac{3}{2},$$

by using $0 \leq L(o,\theta) \leq 1$ almost surely. Since $M(\theta)$ is constant for all $\theta \in \Theta$, it is possible to drop the supremum. Combining all the information above we obtain

$$\begin{aligned}
ER(\theta_{\hat{q}}(P^0_{n,S})) &\leq (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + (1+\delta)\frac{16}{n_1}\log(1+k)\left[1+\frac{3}{2}\frac{1+\delta}{\delta}\right] \\
&= (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + \log(1+k)\frac{3+5\delta}{2\delta}(1+\delta)\frac{16}{n_1} \\
&= (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + \log(1+k)\frac{3+8\delta+5\delta^2}{2\delta}\frac{16}{n_1} \\
&= (1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S})) + \log(1+k)\left(\frac{3}{2\delta}+4+\frac{5}{2}\delta\right)\frac{16}{n_1},
\end{aligned}$$

We can now adjust for the precision in our bound by choosing δ and n. Note that a small delta will mean that the first term $(1+2\delta)ER(\theta_{\tilde{q}}(P^0_{n,S}))$ will become smaller, but this is at the expense that the remainder term becomes larger due to the $\frac{1+\delta}{\delta}$ fraction at the end. By choosing n to be large, we can partially compensate for a smaller delta.
We might, therefore, for each n, choose the δ_n that minimizes the left-hand side for the given n. By substituting $n_1 = n/2$ into the left hand side expression and then expanding it we obtain

$$ER(\theta_{\tilde{q}}(P^0_{n,S})) + 2\delta ER(\theta_{\tilde{q}}(P^0_{n,S})) + \frac{3}{2\delta}\log(1+k)\frac{32}{n} + 4\log(1+k)\frac{32}{n} + \frac{5\delta}{2}\log(1+k)\frac{32}{n},$$

we observe that when n is fixed, two terms remain constant, specifically the first and fourth terms, as they do not depend on δ. The optimal δ_n can be determined as follows:

$$\begin{aligned}
\delta_n &= \arg\min_{\delta} \frac{1}{\delta}\cdot\frac{3\cdot 32}{2n}\log(1+k) + \delta\cdot\left(2ER(\theta_{\tilde{q}}(P^0_{n,S})) + \frac{5\cdot 32}{2n}\log(1+k)\right) \\
&= \arg\min_{\delta} \frac{1}{\delta}\cdot\frac{48}{n}\log(1+k) + \delta\cdot\left(2ER(\theta_{\tilde{q}}(P^0_{n,S})) + \frac{80}{n}\log(1+k)\right) \\
&= \arg\min_{\delta} \frac{1}{\delta}a(n) + \delta b(n),
\end{aligned}$$

Essentially, solving for the minimum is a convex optimization problem, with the terms $a(n) = \frac{48}{n}\log(1+k)$ and $b(n) = 2ER(\theta_{\tilde{q}}(P^0_{n,S})) + \frac{80}{n}\log(1+k)$ remaining constant with respect to n. By differentiating the expression above and setting it equal to zero we obtain

$$0 = \left(\frac{1}{\delta}a(n) + \delta b(n)\right)' = -\frac{1}{\delta^2}a(n) + b(n),$$

and so we obtain the optimum by isolating δ

$$\delta_n = \sqrt{\frac{a(n)}{b(n)}},$$

4 The Ensemble Super Learner

5 Simulation results

6 Discussion

References

[CG16] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[Gyö+02] László Györfi et al. *A distribution-free theory of nonparametric regression*. Vol. 1. Springer, 2002.

[LD03] Mark Laan and Sandrine Dudoit. "Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples". In: *UC Berkeley Division of Biostatistics Working Paper Series* (Jan. 2003).

[VDL06] Aad W. van der Vaart, Sandrine Dudoit, and Mark J. van der Laan. In: *Statistics & Decisions* 24.3 (2006), pp. 351–371. DOI: doi:10.1524/stnd.2006.24.3.351. URL: https://doi.org/10.1524/stnd.2006.24.3.351.