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: May 3, 2023

Contents

1 Introduction

In the context of regression, a natural goal is to estimate a 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 minimized. It turns out that the conditional mean $x \mapsto E(Y \mid X = x)$ minimizes the squared error, but the conditional mean can only be identified if 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 $O(1/n)$. [Lau22][ch. 5]
However, if we are dealing with highly complex data, there is a risk of misspecifying the model by identifying it as an exponential family. Our assumptions may be biased. In such situations, it is more appropriate to utilize non-parametric and data-driven regression methods, such as tree-based algorithms like XGBoost or random forests, to estimate the conditional mean. On the other hand, the assumptions of these data-driven methods regarding $\mathcal{P}$ are not explicit. 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 – regression estimate – 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 the true distribution.
The discrete super learner is then obtained by applying the cross-validation selector to a 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 and notation is similar to [VDL06] and [LD03]: Let a 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)**.** 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

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

A natural aim would be to find the optimal parameter value $\theta^* \in \Theta$ that minimizes the expected L^2-loss, or conditional risk [LD03][p. 2] $R : \theta \to \mathbb{R}$ given by

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

Theorem 2 shows 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*

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

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 $E(Y \mid X = x) = P(Y = 1 \mid X = x)$.
As we do not have access to the data-generating process P, our goal is to estimate θ^*, this means to **learn** the true regression function from our data. We will therefore introduce the terminology **learning algorithm** and **learner** in the context of learning from our data.

2.1 Learning algorithms and learners

Moving forward, we will denote $O_1, \ldots, O_n \in \mathcal{O}$ as our **observations**, and $D_n = (O_1, \ldots, O_n)$ as our **data**.

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. Indeed, it makes sense in this context since we would like to emphasize that the outcome of applying a learning algorithm to our data, $\theta(D_n)$, is a regression function. We refer to that outcome as a **learner**. 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, note that formally $\theta(D_n)$ is a stochastic variable since D_n is stochastic. A learner 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 ω, and subsequently, we can apply our learning algorithm on $D_n(\omega) = (o_1, \ldots, o_n)$, which is a particular instance of a dataset. We will refer the quantity, $\theta(D_n(\omega))$, as a **fitted learner**.

Example 1 (Parametric and nonparametric learning algorithms). An example of a parametric learner is logistic regression. In logistic regression we assume that the conditional probability, $P(Y = 1 \mid X = x)$, can be expressed as $\theta(x) = \text{expit}(\beta x)$ 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. The gradient boosting algorithm, XGBoost, has a number of hyperparameters that can be tuned. These include, number of boosted trees, depth of each tree, learning rates, etc., but most importantly the internal loss objective which could for example be log-loss or mean squared error. XGBoost aims to iteratively refine the fitted learner by approximating the data $x \mapsto f_m(x)$ at each step m. It does so by introducing a new tree $h_m(x)$, which is trained on the error of $f_m(x)$, such that $f_{m+1}(x) = f_m(x) + h_m(x)$. The internal loss of the updated learner, f_{m+1}, evaluated on the training data, is lower than that of the previous learner due to the inclusion of the new tree [CG16].
The parameters of the resulting fit are not directly interpretable. Despite this, XGBoost has demonstrated its ability to model very complex datasets [CG16].

There is a one-to-one correspondence between our data D_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 discrete super learner.

3 The Discrete Super Learner

3.1 Library of learners

We would now like to consider the scenario where we have a set of learning algorithms, $\theta_1, \ldots, \theta_k$. From these algorithms, we can define the **library of learners** $\Theta_k(P_n) = \{\theta_q(P_n) \mid 1 \leq q \leq k\}$ of size k. Our goal is to find $\theta_q(P_n)$, that achieves the lowest risk among our learners – that is to provide an estimate of q for which $q = \arg\min_{1 \leq q \leq k} R(\theta_q(P_n), P)$, we will denote this estimate as $\hat{q}$.

3.2 Cross-validation methodology

To provide $\hat{q}$ 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 $n_1 = \sum_{i=1}^{n} S_i, n_0 = 1 - n_1$ identifies the number of observations in the test and training set respectively.

Example 2 (Random splits). For n observations we have 2^n ways of choosing which observations should be in the training set and in the test 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/test data, since that would also include the combination where $n_1 = 0$. To ensure that there is always a certain amount of observations in our test set, let $n_1 > 0$ be given, we see that there are $\binom{n}{n_1}$ ways of choosing both the test and training set. We can therefore define the distribution of S as

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

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

3.3 Risks and selectors

We now provide the formal definitions for the expected loss associated with our learners. Recall that the expected L^2-loss (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

$$R(\theta, P_n) = \int L(o, \theta) dP_n(o).$$

Given a split variable S for our data D_n, the risk of our learner on the cross-validation test data can be expressed as

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

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

Definition 4 (Expected loss averaged over splits [VDL06])**.** Given data D_n and a split-variable S, the average expected loss for a learner, $\theta_q(P^0_{n,s})$, from a library $\Theta_k(P_n)$ applied to the training data $P^0_{n,S}$ is

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

where P is the data-generating process.

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

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

Definition 5 (Oracle selector)**.** Given the data D_n and a split variable S, the oracle selector depends on our data and is the index of the learner with the lowest averaged expected loss

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

As we are never able to know the oracle, we must proceed via cross-validation to estimate $\tilde{q}$. Cross-validation replaces P with $P^1_{n,S}$ in the second argument of R.

Definition 6 (Cross-validation expected loss)**.** Given data D_n and a split-variable S, the cross-validation expected loss for a learner, $\theta_q(P^0_{n,s})$, from a library $\Theta_k(P_n)$ applied to the training data $P^0_{n,S}$ is

$$E_S R(\theta_q(P^0_{n,S}), P^1_{n,S}),$$

where $P^1_{n,S}$ is the test data as specified by the split-variable.

Definition 7 (Cross-validation selector [LD03])**.** Given the data D_n and a split variable S, the cross validation selector depends on our data and is the index of the learner with the lowest cross-validation expected loss

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

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

Definition 8 (Discrete super learner)**.** The **discrete super learner**, $\theta_{\hat{q}}(P_n)$, created from a library of learners $\Theta_k(P_n) = \{\theta_q(P^0_{n,S}) \mid 1 \leq q \leq k\}$ is the learner chosen by the cross-validation selector applied to D_n

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

Formally, the map above is a random map as P_n is stochastic. Note that in the definition above, $\hat{q}$ depends on our data. The discrete super learner is not one specific leaerner among the learners in the library, but the result of after applying the cross-validation selector to the library.

3.4 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 data or test data, $i = 0$ or $i = 1$.
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*

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

Proof. See appendix □

To control the bounds for the expected loss we introduce Bernstein pairs

Definition 10 (Bernstein pair [VDL06])**.** Given a measurable function $f : O \to \mathbb{R}$, the tuple $(M(f), v(f))$ is a Bernstein pair if

$$M(f)^2 P\left(e^{|f|/M(f)} - 1 - \frac{|f|}{|M(f)|}\right) \leq \frac{1}{2} v(f) \tag{2}$$

Proposition 11. *$(\|f\|_\infty, \frac{3}{2}Pf^2)$ is a Bernstein pair.*

Proof. Following proof is due to [VDL06][ch. 8.1].

$$\begin{aligned}
\|f\|_\infty^2 P\left(e^{|f|/\|f\|_\infty} - 1 - \frac{|f|}{\|f\|_\infty}\right) &= \|f\|_\infty^2 \sum_{k\geq 2}^{\infty} P\frac{|f|^k}{\|f\|_\infty^k k!} = Pf^2 \sum_{k\geq 2}^{\infty} P\frac{|f|^{k-2}}{\|f\|_\infty^{k-2} k!} \\
&\leq Pf^2 \sum_{k\geq 2}^{\infty} \frac{\|f\|_\infty^{k-2}}{\|f\|_\infty^{k-2} k!} = Pf^2 \sum_{k\geq 2}^{\infty} \frac{1}{k!} \\
&= Pf^2(e-2) \leq \frac{3}{4} Pf^2 = \frac{1}{2}\left(\frac{3}{2} Pf^2\right).
\end{aligned}$$

In the first inequality we replace the absolute value of f with the uniform norm, which is larger. we also have □

Lemma 12 (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 $\mathcal{F}$ on $\mathcal{O}$. Then, for any Bernstein pair $(M(f), v(f))$ and for any $\delta > 0$ and $1 \leq p \leq 2$,*

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

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

The expectation above is wrt. the joint probability measure over our observations, $D_n \sim P_O^n = P_{O_1} \otimes P_{O_2} \otimes \cdots \otimes P_{O_n}$.

Theorem 13 (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 $R(\theta, P) \geq 0$ for every $\theta \in \Theta$. Then for $\delta > 0$ and $1 \leq p \leq 2$ it holds that*

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

where k is the number of learners in our library $\{\theta_q(P_{n,S}^0) \mid 1 \leq q \leq k\}$ and S is a split-variable.

In the expectations above, we are taking the expectation wrt. the random split-variable S as well as the joint distribution of our observations. In a more verbose manner one would write

$$E_S E_{D_n} R(\theta_{\hat{q}}(P_{n,S}^0), P) = \int R(\theta_{\hat{q}}(P_{n,S}^0), P) d(P_S \otimes P_O^n).$$

Proof. We will apply lemma 12 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 $R(\theta, P) \geq 0$ for every $\theta \in \Theta$ it follows that $Pf \geq 0$ for every $f \in \mathcal{F}$.
First, we take the expectation wrt. D_n on both sides in lemma 9. For the second term we have

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

Where we use Fubini in the last equality. Recall that $S \perp\!\!\!\perp D_n$, so we can always consider n_1 as fixed given D_n, now applying lemma 12 to the expression above with $n = n_1$ yields

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

Where for the third inequality we take the sup over Θ. We can also bound the third term with the same expression above as lemma 12 is also valid for $-L$. It is now immediate from lemma 9 that

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

The second and third terms above are identical, meaning that they can be combined into one term where the numerator in the first fraction is 16 instead of 8. □

3.5 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 $R(\theta) = \int L(o,\theta)dP(o) \geq 0$ since the loss function is positive. If we plug these numbers in theorem 13, then by using $p = 1$ and

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

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 2. 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 $\theta \in \Theta$:

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

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

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

We see that $a(n)$ converges at a rate of order $1/n$ to 0 and the rate of $b(n)$ depends on the unknown oracle risk. For $n \to \infty$, the risk term $2ER(\theta_{\tilde{q}}(P^0_{n,S}))$ converges to the lowest possible risk achievable by any of the learners, it might be the case that the convergence is fast, i.e. it converges at a rate of order $1/n^{-1/2}$. That is if

$$\delta_n = \sqrt{\frac{a(n)}{b(n)}} = \sqrt{\frac{\frac{48}{n}\log(1+k)}{O(\frac{1}{\sqrt{n}}) + \frac{80}{n}\log(1+k)}} = \sqrt{\frac{48\log(1+k)}{O(\sqrt{n}) + 80\log(1+k)}}$$

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.

[Lau22] Steffen Lauritzen. *Mathematical Statistics: A Concise Course*. 2022.

[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.