



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 15, 2023

Contents

1	Introduction	3
2	Background	3
2.1	Learning algorithms and learners	5
3	The Discrete Super Learner	6
3.1	Library of learners	6
3.2	Cross-validation methodology	6
3.3	Risks and selectors	7
3.4	Oracle inequalities	8
4	The Ensemble Super Learner	12
5	Simulation results	12
5.1	Simulation results for the discrete super learner	13
5.2	Discussion of results	14

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 observations $O = (Y, X)$ 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 $O\left(\frac{1}{\sqrt{n}}\right)$. (Lauritzen 2022)[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 (Vaart, Dudoit, and Laan 2006) and (Laan and Dudoit 2003): 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 $\Theta = \{\theta \mid \theta : \mathcal{X} \rightarrow [0, 1]\}$, 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, $L : \mathcal{O} \times \Theta \rightarrow [0, \infty)$, 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 (Laan and Dudoit 2003)[p. 2] $R : \theta \rightarrow \mathbb{R}$ given by

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

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 $\theta : \mathcal{X} \rightarrow [0, 1]$. 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. Proof emulated from (Györfi et al. 2002)[ch. 1].

Let $\theta \in \Theta$ be arbitrary and $m(x) = E(Y \mid X = x)$, we have

$$\begin{aligned} E|\theta(X) - Y|^2 &= E|\theta(X) - m(X) + m(X) - Y|^2 \\ &= E|\theta(X) - m(X)|^2 + E|m(X) - Y|^2 + 2E[(\theta(X) - m(X))(m(X) - Y)]. \end{aligned}$$

We see that the last term is zero, see that by using the tower rule we have

$$\begin{aligned} E[(\theta(X) - m(X))(m(X) - Y)] &= E[E[(\theta(X) - m(X))(m(X) - Y) \mid X]] \\ &= E[(\theta(X) - m(X))E((m(X) - Y) \mid X)] \\ &= E[(\theta(X) - m(X))(m(X) - E(Y \mid X))] \\ &= E[(\theta(X) - m(X))(m(X) - m(X))] \\ &= 0. \end{aligned}$$

We conclude that

$$\int L(o, \theta) dP(o) = E|\theta(X) - m(X)|^2 + E|m(X) - Y|^2.$$

The first term after the equality is always positive and is 0 only when $\theta = m$, this proves that m minimizes the expression above. \square

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 $O_1, \dots, O_n \in \mathcal{O}$ and $D_n = (O_1, \dots, O_n)$ as our **observations** and **data** respectively.

Definition 3 (Learning algorithm θ). An learning algorithm is a measurable map $\theta : \mathcal{O}^n \rightarrow \Theta$ 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. In practice, we would have observed $O_1 = o_1, \dots, O_n = o_n$, and subsequently, we can apply our learning algorithm on $d_n = (o_1, \dots, o_n)$, which is a particular instance of a dataset. We will refer the quantity, $\theta(d_n)$, 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 (Chen and Guestrin 2016). The parameters of the resulting fit are not directly interpretable. Despite this, XGBoost has demonstrated its ability to model very complex datasets (Chen and Guestrin 2016).

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

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

given that the observations are independent and identically distributed, so permuting the order of the observations does not matter. 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, $\theta_1, \dots, \theta_k$. 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 – from now on referred to as the **split variable** or just **split** – $S^n = (S_1^n, \dots, S_n^n) \in \{0, 1\}^n$ be independent of O_1, \dots, O_n such that $S_i^n = 0$ indicates that O_i should be in the training set and $S_i^n = 1$ indicates that O_i belongs to the test set. Here the superscript n of S^n indicates that the split variable depends on n as in that it is a split of n observations. We can define the empirical distributions over these two subsets, P_{n, S^n}^0 and P_{n, S^n}^1 as

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

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

where $n_1 = \sum_{i=1}^n S_i^n$, $n_0 = 1 - n_1$ 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^n 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^n as

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

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

Example 3 (Cross-validation loss). The test size n_1 is formally a random variable as it depends on S^n . Hold one out cross validation

$$E_{S^n} R(\theta(P_{n,S^n}^0, P_{n,S^n}^1)) = E_{S^n} \frac{1}{n_1} \sum_{i: S_i^n=1} L(o_i, \theta) = E_{S^n} L(o_{S_t}, \theta) = \frac{1}{n} \sum_{i=1}^n L(o_i, \theta)$$

K-fold cross validation

$$E_S R(\theta(P_{n,S}^0, P_{n,S}^1)) = E_S \frac{1}{n_1} \sum_{i: S_i=1} L(o_i, \theta) = \frac{1}{K} \sum_{k=1}^K \sum_{i: s_{k,i}=1} L(o_i, \theta)$$

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^n for our data D_n , the risk of our learner on the cross-validation test data can be expressed as

$$R(\theta, P_{n,S^n}^1) = \int L(o, \theta) dP_{n,S^n}^1(o).$$

The following definitions are analogous to what is stated in section 1 and 2 of (Laan and Dudoit 2003).

Definition 4 (Expected loss averaged over splits (Vaart, Dudoit, and Laan 2006)). Given data D_n and a split-variable S^n , the average expected loss for a learner, $\theta_q(P_{n,s}^0)$, from a library $\Theta_k(P_n)$ applied to the training data P_{n,S^n}^0 is

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

where P is the data-generating process.

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

$$E_{S^n} R(\theta_q(P_{n,S^n}^0), P) = \sum_{s \in \{0,1\}^n} R(\theta_q(P_{n,s}^0), P) \cdot P(S^n = s).$$

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

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

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_{n,S^n}^1 in the second argument of R .

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

$$E_{S^n} R(\theta_q(P_{n,S^n}^0), P_{n,S^n}^1),$$

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

Definition 7 (Cross-validation selector (Laan and Dudoit 2003)). Given the data D_n and a split variable S^n , the cross validation selector depends on our data and is the index of the learner with the lowest cross-validation expected loss

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

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 $\Theta_k(P_n) = \{\theta_q(P_{n,S^n}^0) \mid 1 \leq q \leq k\}$ 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 learner 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, \dots, 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 $G_{n,S^n}^i f = \sqrt{n}(P_{n,S^n}^i - Pf)$ 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 $L : \mathcal{O} \times \Theta \rightarrow \mathbb{R}$ has been given.

Lemma 9 (Lemma 2.1 in (Vaart, Dudoit, and Laan 2006)). *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*

$$\begin{aligned} E_{S^n} \int L(o, \theta_{\hat{q}}(P_{n,S^n}^0)) dP(o) &\leq (1 + 2\delta) E_{S^n} \int L(o, \theta_{\hat{q}}(P_{n,S^n}^0)) dP(o) \\ &\quad + E_{S^n} \frac{1}{\sqrt{n_1}} \max_{1 \leq q \leq k} \int L(o, \theta_q(P_{n,S^n}^0)) d((1 + \delta)G_{n,S^n}^1 - \delta\sqrt{n_1}P)(o) \\ &\quad + E_{S^n} \frac{1}{\sqrt{n_1}} \max_{1 \leq q \leq k} \int -L(o, \theta_q(P_{n,S^n}^0)) d((1 + \delta)G_{n,S^n}^1 + \delta\sqrt{n_1}P)(o) \end{aligned}$$

Proof. See appendix □

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

Definition 10 (Bernstein pair (Vaart, Dudoit, and Laan 2006)). Given a measurable function $f : \mathcal{O} \rightarrow \mathbb{R}$, the tuple $(M(f), v(f))$ is a Bernstein pair if

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

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

Proof. Following proof is due to (Vaart, Dudoit, and Laan 2006)[ch. 8.1].

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

In the first inequality we replace the absolute value of f with the uniform norm, which is larger. \square

Lemma 12 (Lemma 2.2 in (Vaart, Dudoit, and Laan 2006)). 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 (Vaart, Dudoit, and Laan 2006)). For $\theta \in \Theta$ let the numbers $(M(\theta), v(\theta))$ 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

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

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

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

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

Proof. We will apply lemma 12 to the second and third terms on the left hand side of the inequality in lemma 9. Let $\mathcal{F} = \{o \mapsto L(o, \theta_q(P_{n, S^n}^0)) \mid 1 \leq q \leq k\}$, 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

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

Where we use Fubini in the last equality. Recall that $S^n \perp\!\!\!\perp D_n$, 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

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

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

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

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. \square

Remark: One might notice that S^n still appears in the remainder of the bound above, but the only variable in the remainder that depends on S^n is n_1 . It is, therefore, completely possible to drop the expectation wrt. S^n if n_1 is deterministic, for example, if it is a fraction of n .

Corollary 14 (Asymptotic equivalence). *If there exists an $\varepsilon > 0$ such that*

$$E_{D_n} E_{S^n} R(\theta_{\hat{q}}(P_{n, S^n}^0), P) > \varepsilon \quad \text{for all } n \in \mathbb{N},$$

and if $n_1 \rightarrow \infty$ as $n \rightarrow \infty$, then the risk of the super learner is asymptotically equivalent to the risk of the oracle selector, that is

$$\lim_{n \rightarrow \infty} \frac{E_{D_n} E_{S^n} R(\theta_{\hat{q}}(P_{n,S^n}^0), P)}{E_{D_n} E_{S^n} R(\theta_{\tilde{q}}(P_{n,S^n}^0), P)} = 1.$$

For the sake of simplicity and the previous remark we will consider n_1 as a sequence of numbers that increases as n increases, this will allow us to remove E_{S^n} in the remainder term.

Proof. By choosing $p = 2$ in theorem 13, we obtain

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

In the above inequality the remainder term goes to 0 as n_1 increases. Now by dividing the oracle risk on both sides, we obtain

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

As the inequality holds for all $\delta > 0$, we can just set it to 0 (formally one would perhaps choose δ_n decreasing in n). See that by setting $\delta = 0$ the inequality becomes

$$\frac{E_{D_n} E_{S^n} R(\theta_{\hat{q}}(P_{n,S^n}^0), P)}{E_{D_n} E_{S^n} R(\theta_{\tilde{q}}(P_{n,S^n}^0), P)} \leq 1 + \frac{\frac{16}{\sqrt{n_1}} \log(1 + k) \sup_{\theta \in \Theta} \left[\frac{M(\theta)}{\sqrt{n_1}} + \sqrt{v(\theta)} \right]}{E_{D_n} E_{S^n} R(\theta_{\tilde{q}}(P_{n,S^n}^0), P)}$$

The fraction on the right-hand side will converge to 0 because the numerator converges to 0 and the oracle risk is bounded away from 0. We also note that by the definition of the oracle, we have

$$E_{D_n} E_{S^n} R(\theta_{\hat{q}}(P_{n,S^n}^0), P) \leq E_{D_n} E_{S^n} R(\theta_{\tilde{q}}(P_{n,S^n}^0), P),$$

implying that

$$1 \leq \frac{E_{D_n} E_{S^n} R(\theta_{\hat{q}}(P_{n,S^n}^0), P)}{E_{D_n} E_{S^n} R(\theta_{\tilde{q}}(P_{n,S^n}^0), P)},$$

applying the squeeze theorem thus yields the desired result. \square

For the asymptotic equivalence to hold, we must have that $n_1 \rightarrow \infty$ as $n \rightarrow \infty$. This is a reasonable assumption for cross-validation schemes such as K -fold cross-validation or if $n_1 = np$ for $p \in (0, 1)$ where the test size increases with the amount of available data. Note that the condition is not satisfied for leave-one-out cross-validation since n_1 is equal to 1 for every n .

Example 4 (Regression). In a regression context with quadratic loss, the risk for a learner can never be zero unless the binary outcome is deterministic given x . We may begin by noting that the risk of $\theta^*(x) = E(Y | X = x)$ is always positive,

$$\int (Y - E(Y | X))^2 dP \geq 0.$$

Since the integrand is positive, the integral is zero if and only if $Y = E(Y | X)$ almost surely. That is only the case when Y is deterministic given X , i.e. Y is X -measurable. Assuming, therefore, that Y is not deterministic given X , then we note that by the fact that the quadratic loss is strictly proper, then we have for any $\theta \in \Theta$

$$R(\theta, P) \geq R(\theta^*, P) > 0,$$

thus showing that the risk of any learner is strictly positive.

4 The Ensemble Super Learner

5 Simulation results

In the following section we show the results of applying the discrete super learner to a simulated dataset. Our dataset consists of a binary outcome, Y , which depends on two covariates X_1 and X_2 . Our setup is as follows

$$\begin{aligned} X_1 &\sim \text{Unif}(0.5, 15), \\ X_2 | X_1 = x_1 &\sim \mathcal{N}(3.5 - 0.03x_1, 1), \\ Y | X_1 = x_1, X_2 = x_2 &\sim \text{Ber}(\theta_0(x_1, x_2)), \end{aligned}$$

for $\theta_0(x_1, x_2) = \text{expit}(-3.5 - 0.3x_1 + 0.85x_2 + 0.35x_1x_2)$ which is the data-generating regression function. The data-generating regression function is captured by the logistic regression model with interaction terms. We would like to learn θ_0 using the discrete super learner. We will use the following library of learners as an illustrative example:

1. Intercept only logistic regression: $E[Y | X_1, X_2] = \text{expit}(\beta_0)$
2. Logistic regression with main effects: $E[Y | X_1, X_2] = \text{expit}(\beta_0 + \beta_1X_1 + \beta_2X_2)$
3. XGBoost with hyperparameters: `max_depth=3, eta=0.3, n_rounds=100, objective='binary:logistic', booster='dart', nthread=5`

The discrete super learner will use 10-fold cross validation and the internal loss function will be the quadratic loss.

The performance of the discrete super learner will be compared to each individual learner in the library. We show that

1. As the sample size increases, the discrete super learner achieves the minimum risk
2. For a single new observation, the prediction of the discrete super learner on the outcome has the lowest variance

5.1 Simulation results for the discrete super learner

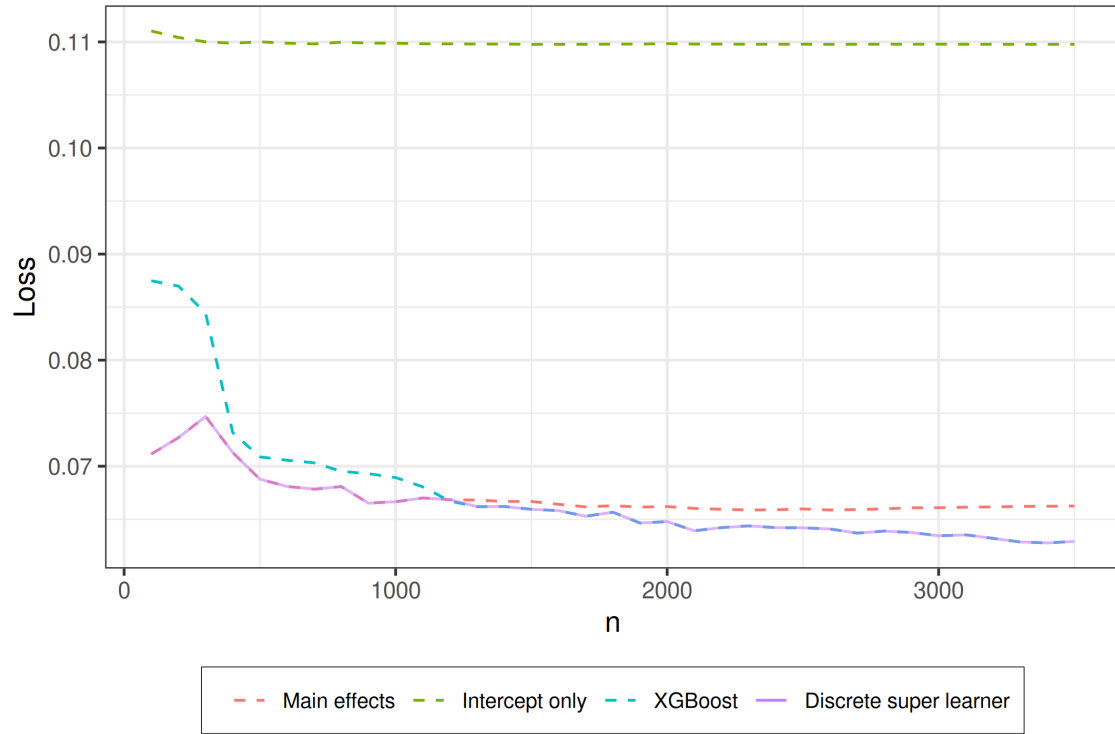


Figure 1: The risk of the discrete super learner compared to other learners. $N = 3500$

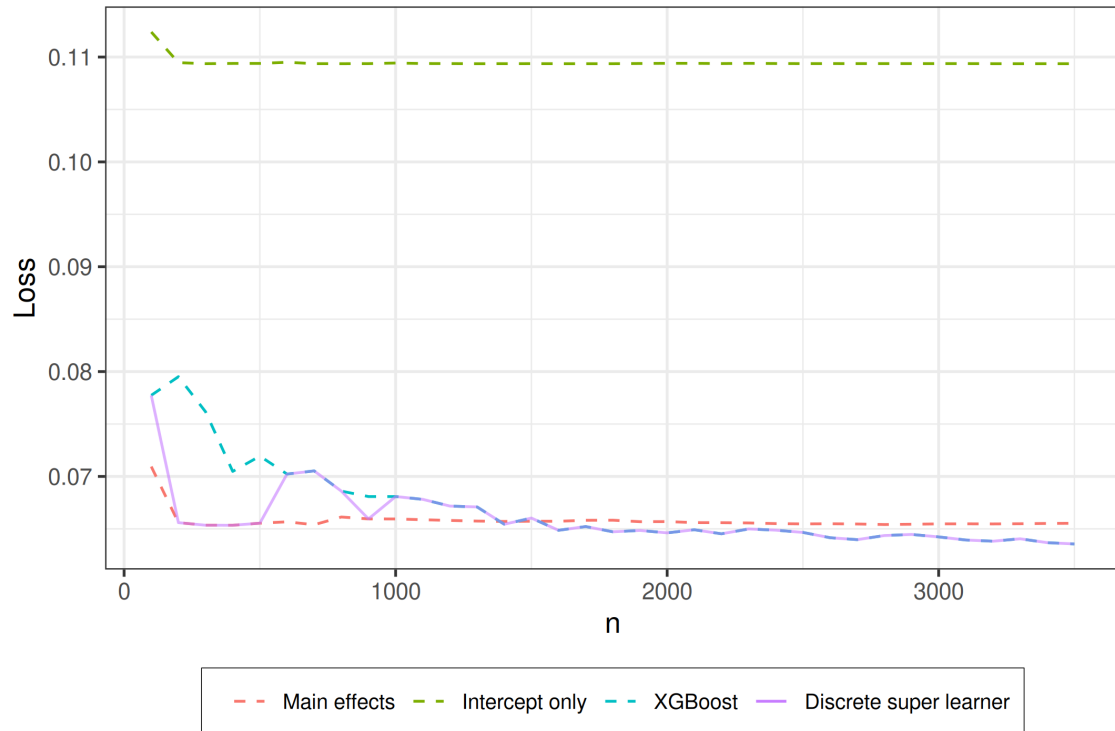


Figure 2: The risk of the discrete super learner compared to other learners. $N = 3500$

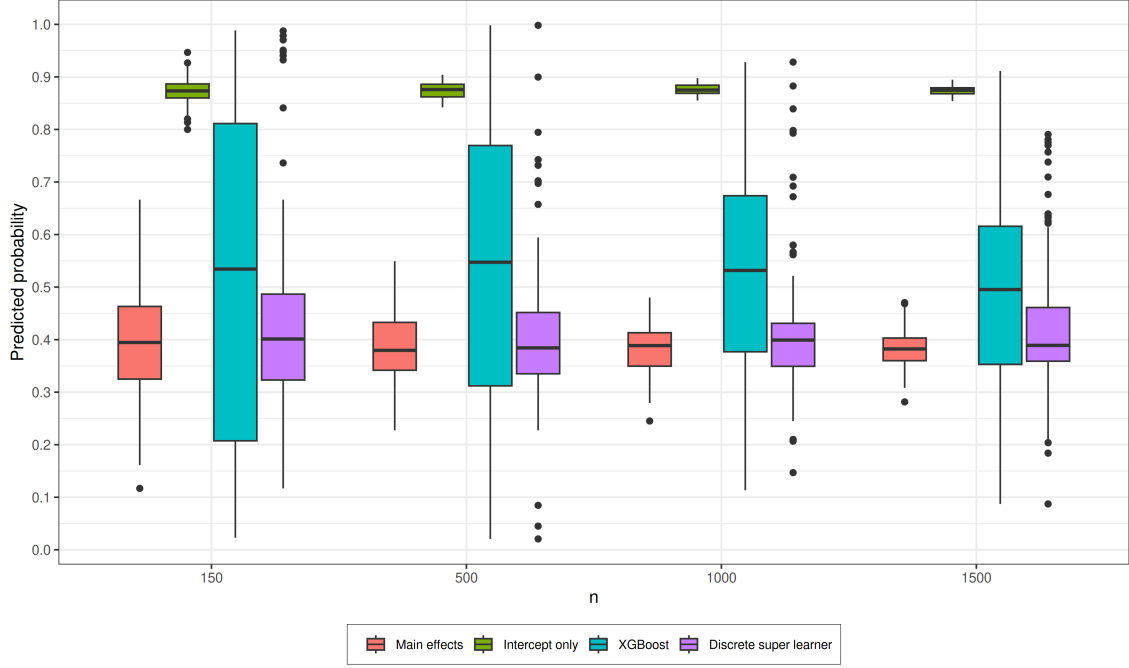


Figure 3: Variances of learner predictions for a single observation, each trained on n samples and evaluated $K = 1000$ times on a single observation

5.2 Discussion of results

Figures 1 and 2 illustrate how the discrete super learner performs in comparison to the learners in terms of loss over number of training samples. The plots are generated for two runs where the model is fitted on $n = 100, 200, \dots, 3500$ observations, as indicated on the x -axis, then the empirical risk is calculated by evaluating each fitted learner on a fixed test sample of size of 5000. The test data is sampled from the data-generating distribution. The first run perfectly illustrates how the discrete super learner is able to achieve the minimum risk. For small training sample sizes, the machine learning method XGBoost has a higher risk than the main effects logistic regression, and it is therefore more desirable for the discrete super learner to choose logistic regression despite the fact that it is misspecified. The discrete super learner consequently achieves the same risk as the logistic regression in the beginning, but for $n > 1200$ the risk of the discrete super learner becomes less than the logistic regression. Here XGBoost begins to achieve a lower risk than the misspecified logistic regression, and so the discrete super learner will choose XGBoost instead.

The second run shows that the discrete super learner might be unable to determine the learner with the lowest risk when the training sample size is small, which results in it moving in a zig-zag pattern between two learners that have quite similar risks. However, we see that the discrete super learner eventually chooses XGBoost as the training sample size becomes larger.

For $n > 1200$ we then observe a shift

In the first run, we observe that the discrete super learner achieves the minimum loss for all sample sizes, the first run is perfectly

References

- [1] László Györfi et al. *A distribution-free theory of nonparametric regression*. Vol. 1. Springer, 2002.
- [2] Mark J. van der 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).
- [3] 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](https://doi.org/10.1524/stnd.2006.24.3.351). URL: <https://doi.org/10.1524/stnd.2006.24.3.351>.
- [4] 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.
- [5] Steffen Lauritzen. *Mathematical Statistics: A Concise Course*. 2022.