

# Super Learners

and their oracle properties

Jinyang Liu (sqf320)

Department of Mathematical Sciences  
University of Copenhagen

June 2023

- 1 Introduction
  - Terminology
  - Cross-validation
- 2 Discrete Super Learner
  - Oracle Property
- 3 Ensemble Super Learner
  - Constrained Regression
- 4 Simulations
  - Validation risk and variance
  - Locally weighted eSL

# Binary regression

Let  $O = (Y, X) \sim P \in \mathcal{P}$  be an observation where  $Y \in \{0,1\}$  and  $X \in \mathcal{X} \subset \mathbb{R}^d$ .

The set of **regression functions**:

$$\Theta = \{\theta \mid \theta : \mathcal{X} \rightarrow [0,1] \text{ measurable}\}.$$

The conditional expectation

$$x \mapsto E(Y \mid X = x) = P(Y = 1 \mid X = x)$$

minimizes the mean squared error (MSE) or risk over  $\Theta$

$$R(\theta, P) = \int (Y - \theta(X))^2 dP.$$

# Terminology

We observe  $D_n = (O_1, \dots, O_n) \in \mathcal{O}^n$ .

## Learning algorithm

A learning algorithm is a measurable map  $\psi : \mathcal{O}^n \rightarrow \Theta$  for  $n \in \mathbb{N}$ .

## Learner or fitted learner

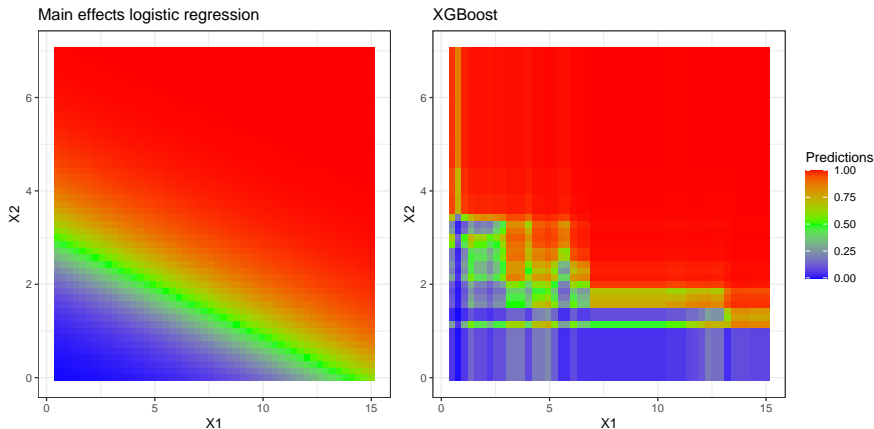
A learner  $\psi(D_n) \in \Theta$  is the outcome of applying  $\psi$  to the data  $D_n$ .

We usually have a **library of learning algorithms**,

$$\Psi = \{\psi_q \mid 1 \leq q \leq k\},$$

which we can use to estimate  $E(Y|X = x)$ .

# Two learners: Logistic Regression and XGBoost



**Figure:** The predictions of the main effects logistic regression and XGBoost fitted on 1,000 observations.

# K-fold Cross-validation

There is a one-to-one correspondence between our data  $D_n$  and the empirical measures over  $n$  observations

$$P_n = \sum_{i=1}^n \delta_{O_i},$$

$K$ -fold cross-validation splits  $D_n$  into **validation** and **training** sets. The validation sets are indexed by  $s \in \{1, \dots, K\}$  and we denote the empirical measure over the validation set  $s$  as

$$P_{n,s}^1 := \frac{1}{n_1} \sum_{i:s(i)=s} \delta_{O_i}, \quad P_{n,s}^0 := \frac{1}{n_0} \sum_{i:s(i) \neq s} \delta_{O_i}.$$

Here  $s(i)$  denotes whether  $O_i$  is in the validation set  $s$ , and  $n_1, n_0$  are the number of observations in the validation and training sets respectively.

## The **super learner** $K$ -fold cross-validation procedure

- 1 Randomly split  $D_n$  into  $K$  disjoint and exhaustive validation sets
- 2 For each  $s \in \{1, \dots, K\}$  fit each  $\psi \in \Psi$  on the training data  $P_{n,s}^0$  and obtain  $\psi(P_{n,s}^0)$
- 3 For each  $s$  and  $\psi$ , use  $\psi(P_{n,s}^0)$  to predict on the validation set to obtain **level-1 covariates**:

$$Z_i = \left( \psi_1(P_{n,s(i)}^0)(X_i), \dots, \psi_k(P_{n,s(i)}^0)(X_i) \right)$$

- 4 Calculate the MSE of  $\psi$  on the validation sets for  $s \in \{1, \dots, K\}$

$$R(\psi(P_{n,s}^0), P_{n,s}^1) = \frac{1}{n_1} \sum_{i:s(i)=s} (Y_i - \psi(P_{n,s}^0)(X_i))^2$$

# Overview

- 1 Introduction
  - Terminology
  - Cross-validation
- 2 Discrete Super Learner
  - Oracle Property
- 3 Ensemble Super Learner
  - Constrained Regression
- 4 Simulations
  - Validation risk and variance
  - Locally weighted eSL



# Discrete Super Learner

Cross-validation allows us to select the algorithm with the lowest **empirical risk**

$$\hat{\psi}_n := \arg \min_{\psi \in \Psi} \frac{1}{K} \sum_{s=1}^K R(\psi(P_{n,s}^0), P_{n,s}^1),$$

also known as the **cross-validation selected algorithm**. The **discrete super learner** is simply the cross-validation selected algorithm fitted on the entire dataset

$$X \mapsto \hat{\psi}_n(P_n)(X).$$

It is compared to the **oracle selected learning algorithm** that has the true minimum risk

$$\tilde{\psi}_n := \arg \min_{\psi \in \Psi} \frac{1}{K} \sum_{s=1}^K R(\psi(P_{n,s}^0), P).$$

## Theorem (Asymptotic equivalence)

*If there exists an  $\varepsilon > 0$  such that*

$$E_{D_n} \frac{1}{K} \sum_{s=1}^K R(\tilde{\psi}_n(P_{n,s}^0), P) > \varepsilon \quad \text{for all } n \in \mathbb{N},$$

*and if  $n_1 = f(n)$  for some polynomial function  $f$ , then the risk of the super learner is asymptotically equivalent with the risk of the oracle selected learner, that is*

$$\lim_{n \rightarrow \infty} \frac{E_{D_n} \frac{1}{K} \sum_{s=1}^K R(\hat{\psi}_n(P_{n,s}^0), P)}{E_{D_n} \frac{1}{K} \sum_{s=1}^K R(\tilde{\psi}_n(P_{n,s}^0), P)} = 1.$$

# Overview

- 1 Introduction
  - Terminology
  - Cross-validation
- 2 Discrete Super Learner
  - Oracle Property
- 3 Ensemble Super Learner
  - Constrained Regression
- 4 Simulations
  - Validation risk and variance
  - Locally weighted eSL

# Ensemble Super Learner

## Definition (Level 1 data)

The *level 1 data*, combines the observed outcomes  $Y_i$  with the level 1 covariates:

$$\mathcal{L}_n = \{(Y_i; \psi_1(P_{n,s(i)}^0)(X_i), \dots, \psi_k(P_{n,s(i)}^0)(X_i))\}_{i=1}^n.$$

## Definition (Meta learning algorithm)

A *meta learning algorithm*  $\Phi$  is a learning algorithm that uses the level 1 data to create a learner:  $\mathcal{L}_n \mapsto \Phi(\mathcal{L}_n)$ .

# Ensemble super learner

## Definition

Let  $\phi = \Phi(\mathcal{L}_n)$  be the outcome of applying a meta learning algorithm  $\Phi$  to the level 1 data, then the map

$$x \mapsto \phi(\psi_1(P_n)(x), \dots, \psi_k(P_n)(x)),$$

is called the *ensemble super learner*.

Each learning algorithm is fitted on the entire dataset, and a meta learner is used to combine the predictions.

# Ensemble Super Learner: Constrained Regression

There are many possible meta learning algorithms. E.g., one can fit a weighted linear combination of the learning algorithms where

$$\phi_a(z) = a \cdot z, \quad \sum_{q=1}^k a_q = 1, a_q \geq 0 \text{ for all } q,$$

such that

$$\phi_a(\psi_1(P_n)(x), \dots, \psi_k(P_n)(x)) = \sum_{q=1}^k a_q \psi_q(P_n)(x) \in [0,1].$$

The optimal weighs,  $a$ , can be found by solving a **constrained least squares** on the level 1 data using quadratic programming techniques.

# Overview

- 1 Introduction
  - Terminology
  - Cross-validation
- 2 Discrete Super Learner
  - Oracle Property
- 3 Ensemble Super Learner
  - Constrained Regression
- 4 Simulations
  - Validation risk and variance
  - Locally weighted eSL

We consider a simulated dataset consisting of two covariates  $X_1, X_2$  and a binary outcome  $Y \in \{0,1\}$

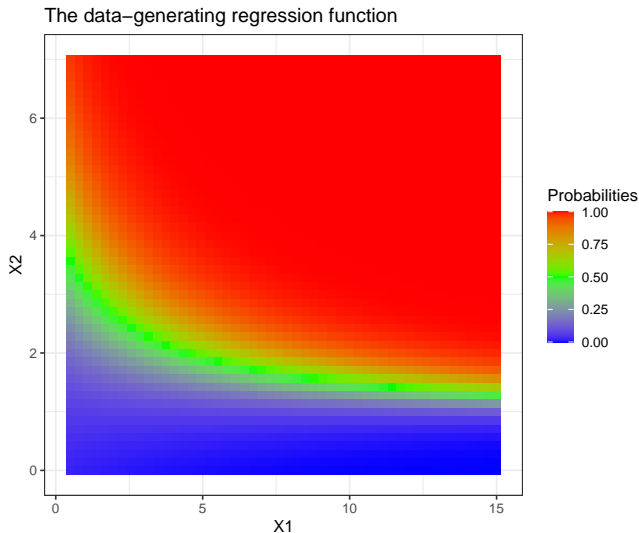
$$X_1 \sim \text{Unif}(0.5, 15),$$

$$X_2 \mid X_1 = x_1 \sim \mathcal{N}(3.5 - 0.03x_1, 1),$$

$$Y \mid X_1 = x_1, X_2 = x_2 \sim \text{Ber}(\theta_0(x_1, x_2)),$$

where  $\theta_0(x_1, x_2) = \text{expit}(-3.5 - 0.3x_1 + 0.85x_2 + 0.35x_1x_2)$ .





**Figure:** The data-generating regression plotted as a heat map. The two covariates  $X_1$  and  $X_2$  are mapped by the regression function  $\theta_0$  to a probability as indicated by the colors.

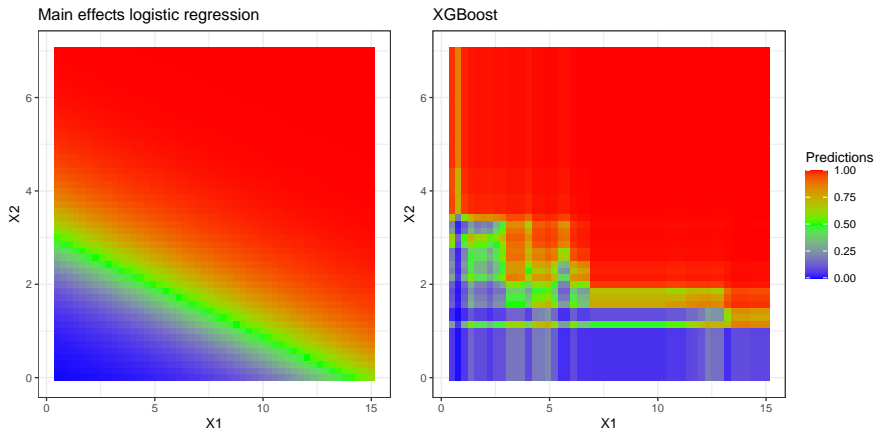
# Simulations: Library of Algorithms

The regression is captured by using logistic regression with interaction terms. We use the following library of learning algorithms:

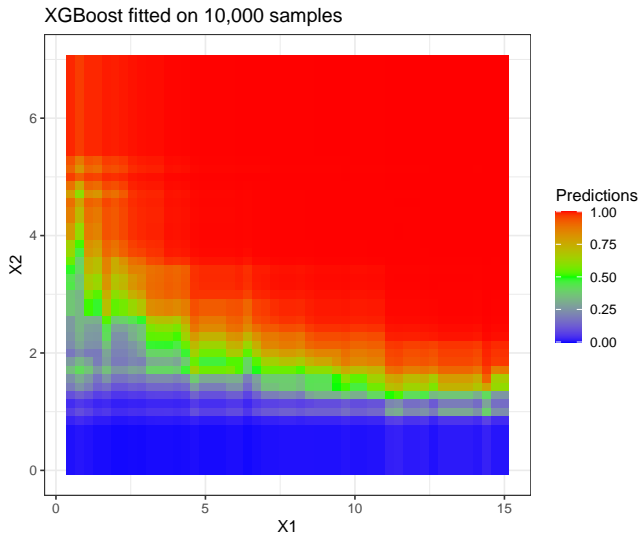
- 1 Intercept only logistic regression:  $E[Y | X_1, X_2] = \text{expit}(\beta_0)$
- 2 Logistic regression with main effects:  
 $E[Y | X_1, X_2] = \text{expit}(\beta_0 + \beta_1 X_1 + \beta_2 X_2)$
- 3 XGBoost with hyperparameters: `max_depth=3, eta=0.3, n_rounds=100, objective='binary:logistic', booster='dart', nthread=5`

The intercept model is included as a **benchmark**, no learning algorithm should perform worse than the benchmark.

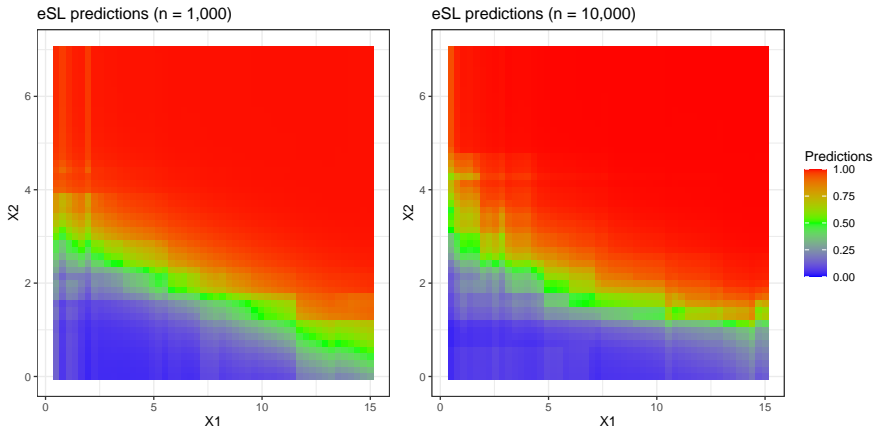
# Simulations: Logistic Regression and XGBoost



**Figure:** The predictions of the main effects logistic regression and XGBoost fitted on 1,000 observations.

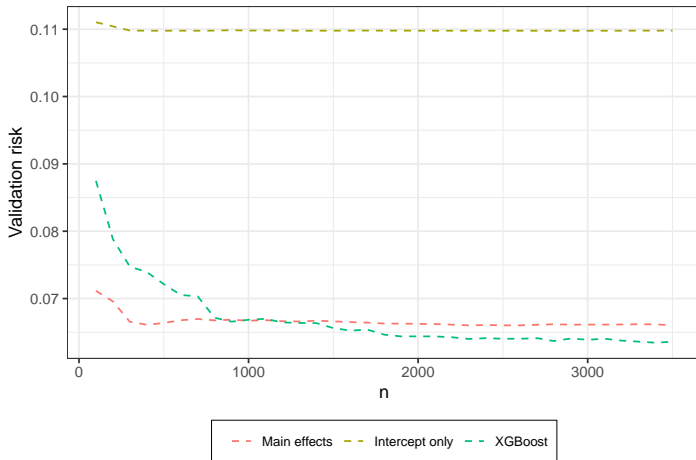


**Figure:** XGBoost becoming better at approximating the regression as the sample size increases. Here the predictions of XGBoost are visualized for a training sample size of 10,000.



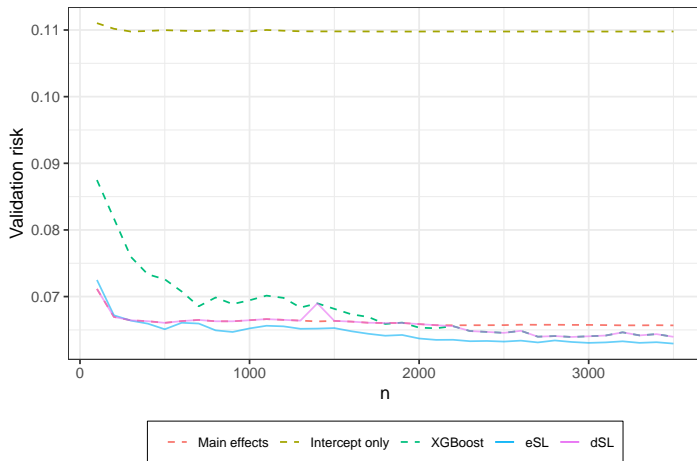
**Figure:** The predictions by the ensemble super learner using the constrained regression meta learning algorithm fitted on 1,000 and 10,000 observations.

# Simulations: Risk and Variance over Training Samples

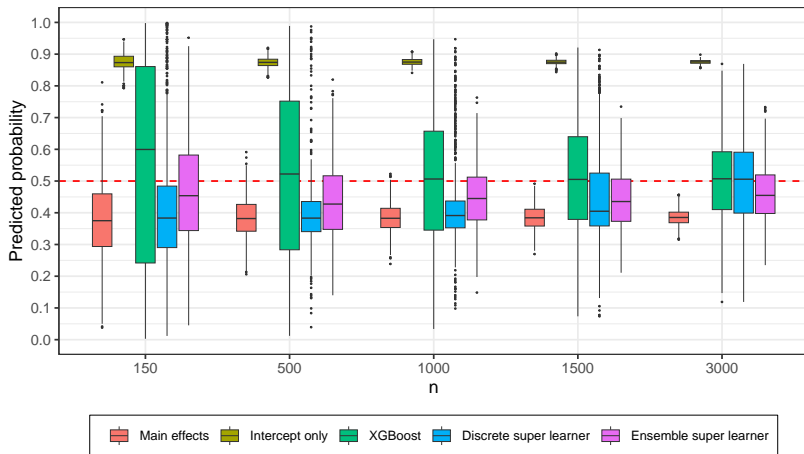


**Figure:** The validation risk of the learning algorithms where training samples are  $n = 100, 200, \dots, N = 3,500$ .

# Simulations: Risk and Variance over Training Samples



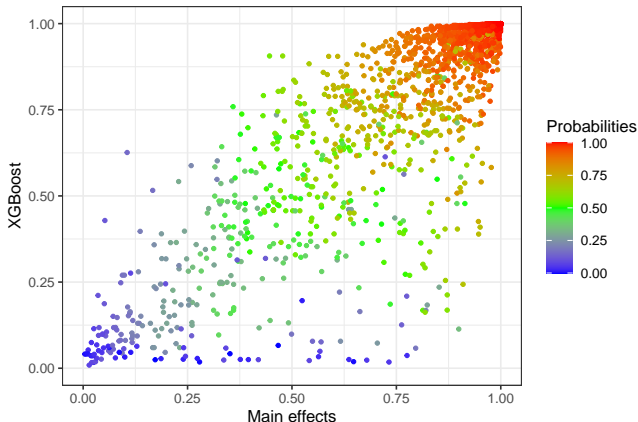
**Figure:** The validation risk of the super learners compared to the library of algorithms where the number of training samples are  $n = 100, 200, \dots, N = 3,500$ .



**Figure:** The variance of the super learners compared to other algorithms. Each algorithm is fitted  $K = 1,000$  times on  $n$  samples and used to predict  $K$  times on a single observation.



# Locally Weighted Ensemble Super Learner



**Figure:** Predictions of XGBoost vs main effects model. The learners' prediction on a single observation represent a point in the unit square, the point is colored by the probability obtained from applying the true regression function on that observation.

# Clustering level 1 covariates using $k$ -means

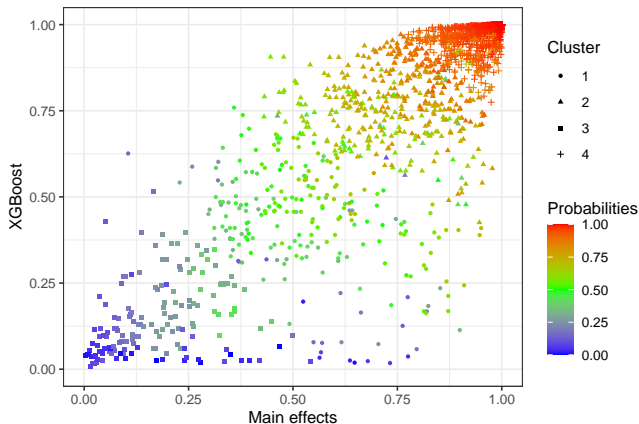
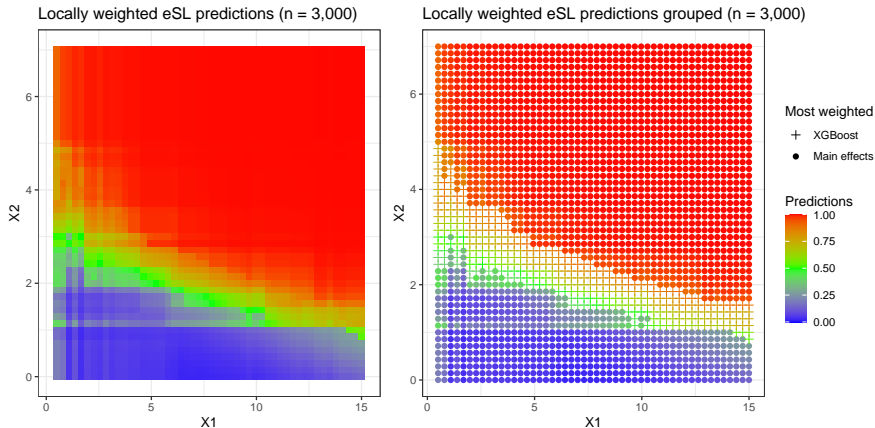


Figure: The predictions are clustered into 4 groups using  $k$ -means clustering.

# Locally Weighted Ensemble Super Learner



**Figure:** Predictions of locally weighted ensemble super learner, where the predictions of XGBoost have highest weight when  $p \approx 0.5$

# Concluding Remarks

- ① Similar asymptotic results hold for the ensemble super learner
- ② The super learner seeks to optimize the bias-variance tradeoff