Mathematic formula for SuperLearner Algorithm

Yehu Chen

January 25, 2018

1 INTRODUCTION

Super learning is a general loss based learning method that has been proposed and analyzed theoretically in van der Laan et al. (2007). It is a prediction method designed to find the optimal combination of a collection of prediction algorithms. The super learner algorithm finds the combination of algorithms minimizing the cross-validated risk. The super learner framework is built on the theory of cross-validation and allows for a general class of prediction algorithms to be considered for the ensemble.

2 ALORITHM

Observe the learning data set $X_i = (Y_i, W_i), i = 1, ..., n$ where Y is the outcome of interest and W is a p-dimensional set of covariates. The objective is to estimate the function $f(W) = E(Y|W)$. The function can be expressed as the minimizer of the expected loss:

$$J = \sum_{i=1}^{n} (Y_i - f(W_i))^2$$

$$f = \underset{f}{\operatorname{argmin}}(J)$$

Actually, for each algorithm L, we can find one corresponding f minimizing the loss function. Suppose we have a library of algorithm $\mathscr{L}$, where the cardinality of $\mathscr{L}$ is K.

1. Fit each algorithm in L on the entire data set $X = X_i : i = 1, ..., n$ to estimate $f_k(W_i), k = 1, ..., K$.

2. Split the data set X into a training and validation sample, according to a $V-$fold cross- validation scheme: splits the ordered n observations into V -equal size groups, let the $v-$th group be the validation sample, and the remaining group the training sample, $v = 1, ..., V$. Define $T()$ to be the th training data split and $V(v)$ to be the corresponding validation data split. $T() = X\ V(v), v = 1, ..., V$.

3. For the v th fold, fit each algorithm in $\mathscr{L}$ on $T(v)$ and save the predictions on the corresponding validation data, $f_{k,T(v)}(W_i), X_i \in V(v)$ for $v = 1,...,V$.

4. Stack the predictions from each algorithm together to create a n by K matrix, $Z_{i,k} = f_{k,i}, i = 1,...,n,\ k = 1,...,K$.

5. Propose a family of weighted combinations of the candidate estimators indexed by weight- vector α:

$$m(z_i|\alpha) = \sum_{k=1}^{K} \alpha_k J_k,\ \sum_{k=1}^{K} \alpha_k = 1$$

6. Determine the α that minimizes the cross-validated risk of the candidate estimator over all allowed -combinations to create the final super learner fit.

3 MATH

3.1 Symbols

1. Denote Y as a n by 1 matrix that is the actual respond vector.

2. Denote A as a n by k matrix that is the cross-validation matrix. Each column of matrix A is the estimator vector for one algorithm in our algorithm library $\mathscr{L}$.

3. Denote α as n by 1 matrix that is the coefficient vector for SuperLearner algorithm.

3.2 Matrix representation of SuperLearner Loss function

α is chosen as n by 1 matrix that minimizes error for SuperLearner algorithm. We choose Loss function as followed:

$$\text{Loss}(\alpha) = \sum_{i=0}^{n} ||Y_i - m(z_i|\alpha)||^2$$

We take trace norm for the vector, and loss function would be

$$\text{Loss}(\alpha) = tr(Y - A\alpha)^T (Y - A\alpha)$$

3.3 Matrix representation of α

We want to minimize the loss function under the constraint

$$\sum_{k=1}^{K} \alpha_k = 1$$

To the Lagrange multiplier rule, we define a new function

$$F(\lambda, \alpha) = tr(Y - A\alpha)^T (Y - A\alpha) + \lambda(\sum_{k=1}^{K} \alpha_k - 1)$$

We first find the derivative of the loss function

$$\begin{aligned}\nabla_\alpha \text{Loss}(\alpha) &= \nabla_\alpha tr(Y - A\alpha)^T (Y - A\alpha) \\ &= \nabla_\alpha tr(Y^T Y - \alpha^T A^T Y - Y^T A\alpha + \alpha A^T A\alpha \\ &= \nabla_\alpha tr(-2Y^T A\alpha + \alpha A^T A\alpha \\ &= -2A^T Y + 2A^T A\alpha\end{aligned}$$

so the derivative of function F would be

$$\frac{\partial F}{\partial \alpha_k} = -2A^T Y_k + 2A^T A e_k \alpha + \lambda$$

$$\frac{\partial F}{\partial \lambda} = \sum_{k=1}^{K} \alpha_k - 1$$

where e_k is the k-th standard normal base. Leaving them to zero, we have

$$2A^T A e_k \alpha_k + \lambda = 2A^T Y_k$$

$$\sum_{k=1}^{K} \alpha_k = 1$$

To see in matrix

$$\begin{pmatrix} 2A^T Ae_1 & & 0 & 1 \\ & \ddots & & 1 \\ 0 & & 2A^T Ae_k & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \lambda \end{pmatrix} = \begin{pmatrix} 2A^T Y_1 \\ \vdots \\ 2A^T Y_k \\ 1 \end{pmatrix}$$

so take the inverse, we will get α

$$2A^T A e_k \alpha_k + \lambda = 2A^T Y_k$$

$$\sum_{k=1}^{K} \alpha_k = 1$$

To see in matrix

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \lambda \end{pmatrix} = \begin{pmatrix} 2A^T Ae_1 & & 0 & 1 \\ & \ddots & & 1 \\ 0 & & 2A^T Ae_k & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 2A^T Y_1 \\ \vdots \\ 2A^T Y_k \\ 1 \end{pmatrix}$$

4 Alternative Norm for Constraint extrema

Polley and van der Laan used L_1 norm for constraining α. Since the dimension of the sample space is finite, we rewrite the constrain in L_2 norm.

4.1 Matrix representation of α

We want to minimize the loss function under the constraint

$$||\alpha||_2 = 1$$

$$\sum_{k=1}^{K} \alpha_k^2 = 1$$

To the Lagrange multiplier rule, we define a new function

$$F(\lambda, \alpha) = tr(Y - A\alpha)^T (Y - A\alpha) + \lambda (\sum_{k=1}^{K} \alpha_k^2 - 1)$$

the derivative of function F would be

$$\frac{\partial F}{\partial \alpha} = -2A^T Y + 2A^T A\alpha + \lambda$$

$$\frac{\partial F}{\partial \lambda} = \sum_{k=1}^{K} \alpha_k^2 - 1$$

Leaving them to zero, we have

$$\alpha = (A^T A + \lambda I_k)^{-1} A^T Y$$

$$\sum_{k=1}^{K} \alpha_k^2 = 1$$

5 CONCLUSIONS

Although R has SuperLearner package, it is important to understand the algorithm. However, closed form for SuperLearner coefficient requires calculating extrema under constraints, leaving the formula not elegant. For the purpose of the project, the R package would be enough.

References

[1] Polley, Eric C. and van der Laan, Mark J., "Super Learner In Prediction" (May 2010). U.C. Berkeley Division of Biostatistics Working Paper Series. Working Paper 266. http://biostats.bepress.com/ucbbiostat/paper266