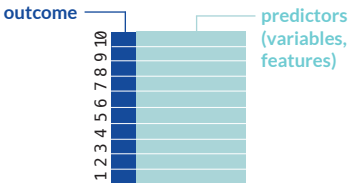
A VISUAL GUIDE TO SUPERLEARNING

Katherine Hoffman, MS
@rkatlady

Superlearning, or stacking, **weights** the results of **many individual statistical learning algorithms** to create an optimal overall prediction algorithm. Superlearner predictions are guaranteed to perform at least as well as any of the individual learners in large sample sizes.

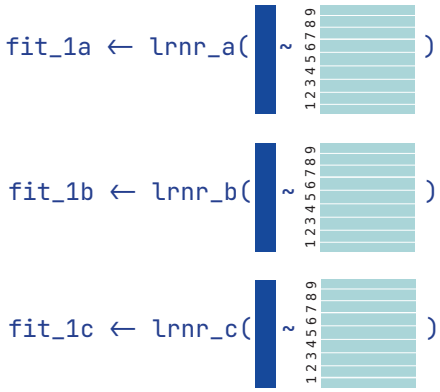
STEP 1

Split data into 10 blocks in preparation for 10-fold cross validation.



STEP 2

Train multiple base learners on 9 of the 10 blocks of data.

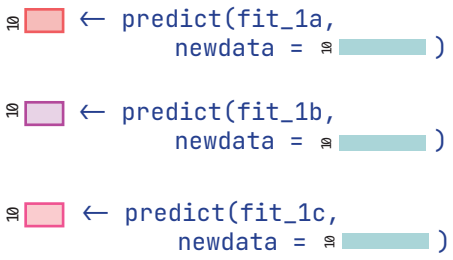


Base learners can include any number of parametric or non-parametric supervised statistical learning algorithms.

An example of three base learners for a binary variable could be random forest, gradient boosting, and logistic regression.

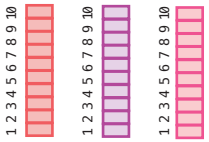
STEP 3

Obtain predictions from each base learner for the held-out block of data.



STEP 4

Repeat until each of the 10 blocks have served as the hold-out data and you have three sets of cross-validated predictions spanning the full data set.



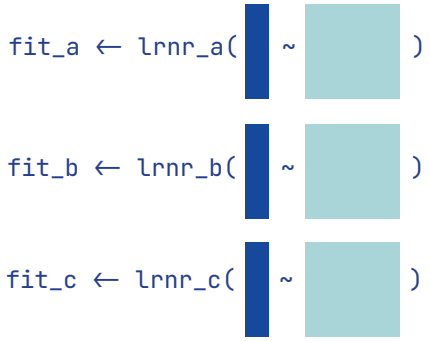
STEP 5

Using a new learner, a metalearner, predict the outcome using the three sets of cross-validated predictions.

The metalearner can be as simple as a generalized linear model. As with any statistical learning algorithm, the choice reflects a loss function we want to minimize.

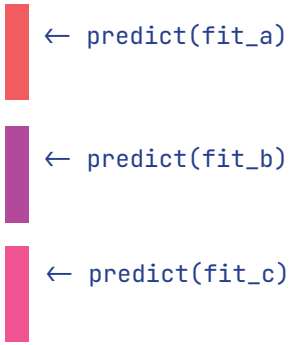
STEP 6

Fit the base learners on the entire data set.



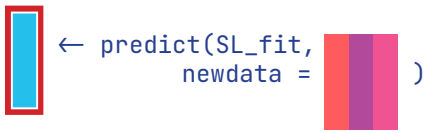
STEP 7

Obtain predictions from the full data set for each learner.



STEP 8

Use the coefficients from Step 5 to weight the full data predictions from Step 7. These are the final superlearner predictions.



The final superlearner predictions are a weighted combination of the base learners' predictions.

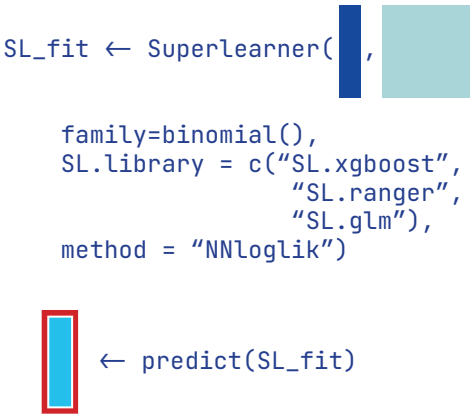
EVALUATION

To test the prediction cabability of the superlearner algorithm and prevent overfitting, the entire algorithm (Steps 1-8) could be cross-validated.

APPLICATION

There are several R packages to implement superlearning including `SuperLearner`, `sl3`, `h2o`, and `ml3`.

The following example uses the `SuperLearner` package to predict a binary outcome with three learners: gradient boosting (`xgboost`), random forest (`ranger`), and logistic regression (`glm`) with a loss function/ metalearning step of negative log-likelihood (`method="NNloglik"`):



REFERENCES

The algorithm described here is detailed in *Targeted Learning, Chapter 3: Superlearning* by Eric Polley, Sherri Rose, and Mark van der Laan.

A full tutorial with more references, R code, and explanations can be found at: www.khstats.com/blog/sl/superlearning.