Gradient Boosting

Introduction to Statistical Learning Section 8.2.3

Recall: Ensemble Methods

Recall that in machine learning, an **ensemble** method is one which combines the results of many (usually simpler) predictors to generate one final prediction.

Random Forests are an example of an ensemble, because a random forest trains a large number of decision trees and combines the predictions to create a final prediction.

Random forests uses an ensemble method called **bootstrap aggregation (bagging)**, because the individual decision trees are trained on bootstrap samples of the training set, and the final predictions are obtained by aggregating the predictions.

Recall: Ensemble Methods

We will look at another ensemble method, called **boosting**.

For bagging, our decision trees are built “in parallel” and predictions are aggregated at the end.

On the other hand, the decision trees for boosting are trained in sequence.

Big Idea: Each subsequent learner tries to correct the mistakes of the prior ones.

Boosting

Boosting Algorithm (for Regression):

1. Fit an initial model to the data: $f_0(\vec{x})$
2. Generate predictions from the model and calculate the residuals:

$$r_i = y_i - f_0(\vec{x}_i)$$

3. Train a new model to predict the residuals: $f_1(x)$
4. Update your model using (a shrunken version of) this new model:

$$f_0(x) \leftarrow f_0(x) + \lambda \cdot f_1(x)$$

5. GOTO 2

Boosting Example

Let’s say we are building a model to predict home prices. We have a very small training set of only 4 values, but decide to proceed anyway. We train our initial model $f_0(x)$ (say a simple decision tree model)

True Value y_i	Predicted Value $f_0(x_i)$	Residual
$190,000	$250,000	
$215,000	$205,000	
$280,000	$305,000	
$320,000	$275,000	

Boosting Example

Now, we can see how far off we were. That is, we can find the residuals.

True Value y_i	**Predicted Value** $f_0(x_i)$	**Residual** $r_i = y_i - f_0(x_i)$
$190,000	$250,000	-$60,000
$215,000	$205,000	$10,000
$280,000	$305,000	-$25,000
$320,000	$275,000	$45,000

Boosting Example

Now, we build a new model, $f_1(x)$, to try and predict the residuals:

Residual r_i	Predicted Residual $f_1(x_i)$
-$60,000	-$48,000
$10,000	$9,000
-$25,000	-$28,000
$45,000	$49,000

Boosting Example

Combine f_0 and f_1 to get our new predicted values:

True Value y_i	Initial Predicted Value $f_0(x_i)$	Predicted Residual $f_1(x_i)$	Final Predicted Value $f_0(x_i) := f_0(x_i) + f_1(x_i)$
\$190,000	\$250,000	-\$48,000	
\$215,000	\$205,000	\$9,000	
\$280,000	\$305,000	-\$28,000	
\$320,000	\$275,000	\$49,000	

Boosting Example

Combine f_0 and f_1 to get our new predicted values:

True Value y_i	Initial Predicted Value $f_0(x_i)$	Predicted Residual $f_1(x_i)$	Final Predicted Value $f_0(x_i) := f_0(x_i) + f_1(x_i)$
\$190,000	\$250,000	-\$48,000	\$202,000
\$215,000	\$205,000	\$9,000	\$214,000
\$280,000	\$305,000	-\$28,000	\$277,000
\$320,000	\$275,000	\$49,000	\$324,000

Boosting Example

Now, we can use our new $f_0(x)$ to compute the new residuals and repeat the process.

True Value y_i	Predicted Value $f_0(x_i)$	Residual $r_i = y_i - f_0(x_i)$
$190,000	$202,000	
$215,000	$214,000	
$280,000	$277,000	
$320,000	$324,000	

Boosting Example

Now, we can use our new $f_0(x)$ to compute the new residuals and repeat the process.

True Value y_i	**Predicted Value** $f_0(x_i)$	**Residual** $r_i = y_i - f_0(x_i)$
\$190,000	\$202,000	-\$12,000
\$215,000	\$214,000	\$1,000
\$280,000	\$277,000	\$3,000
\$320,000	\$324,000	-\$4,000

Boosting

Why do we need λ?

It acts as a form of regularization, helping prevent overfitting. Instead of taking the whole new prediction, we take only a fraction of it.

Small values of lambda (0.01 or below) can help avoid overfitting, at the expense of increased training time.

$$f_0(x) \leftarrow f_0(x) + \boxed{\lambda} \cdot f_1(x)$$

Boosting Example

Note: In reality, when updating f$_0$(x), we will “shrink” our predictions for the residuals by a “learning rate” **λ**, which tends to improve the generalizability of the model overall, at the cost of increasing computation time.

$$f_0(x) \leftarrow f_0(x) + \lambda \cdot f_1(x)$$

Boosting

Other possible modifications:

Take only a fraction of the training samples at each stage. This decreases the variance of the predictions.

Boosting - Feature Importances

Gradient boosted trees are able to indicate feature importances - average (over all trees in the ensemble) the reduction in RMSE when that feature is used to split the data.

Difference Between Boosting and Random Forests

Random Forest:

- Trees are trained independently.
- Final predictions are obtained by averaging the predictions from all trees.

Boosting:

- Trees are trained in sequence.
- Final predictions are obtained by taking a weighted sum of individual trees.

XGBoost

We will be working with a specific implementation of gradient boosting, called **XGBoost**, which was released in 2014 and gained popularity after winning the Higgs Machine Learning Challenge, where the goal was to classify signals generated by particle colliders.

In addition to the basic gradient boosting algorithm, it adds some additional tweaks, leading to stronger overall performance:

- Regularization (think LASSO or Ridge Regression)
- Better search algorithm to train component decision trees
- Structure allowing for parallel computing for faster training