# Boosting

# Idea behind Boosting

**Main idea:** Try to go from underfitting to overfitting by taking small steps.

**General steps:**

- Use a weak base learner (boosting stumps, OLS with one predictor)
- Improve on the base learner using the residuals
- Repeat

# Gradient Boosting: Fitting Residuals

Treat **residuals as the outcome**, with the same predictors.

After fitting a model to predict residuals, **combine original model + residual model** to get updated model.

Get residuals from that and **repeat**.

- *See lecture slides on Gradient Boosting

# Weak Learner

We use a **weak learner** because we are trying to **slowly** improve our model.

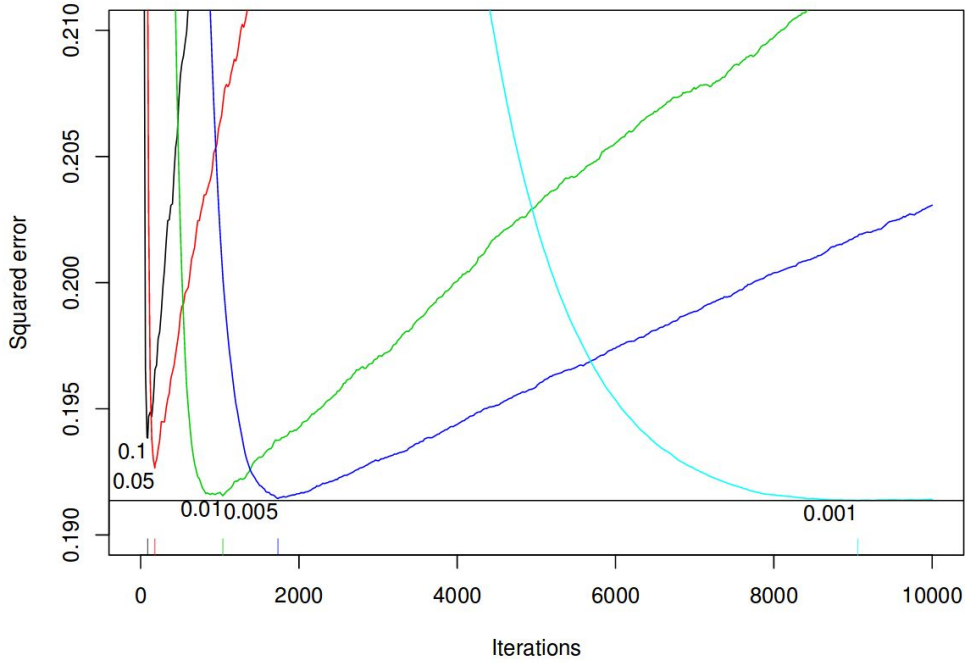
Each subsequent iteration in boosting makes the model **slightly more complex**.

We don’t want to make too big of a jump, because we will just end up overfitting then.

# Shrinkage and Tuning

Shrinkage parameter adjust how big of a step you take

- Think of this as how much the residual model updates the overall model.
- Lamba similar to elastic net, but applied differently



# Grid Search and Random Search

**Remember**: goal is to find the combination of tuning parameters that **minimizes test error.**

If there are lots of combinations of tuning parameters, it may be better to use **random search** to lessen the computation time.

However, this is just a **starting point**. Tuning is an iterative process!

# Interpretability

Boosting methods **suffer from interpretability** because of the approach.

- What does a model on the residuals mean in the long run? What about over 100 iterations?

**Model-based boosting** aims to get some of that back by creating a linear form at the end.

- *See lecture slides on model-based boosting

# Computation Time

**Note:** For boosting methods, the current iteration relies on the previous iteration

- Because you need to fit a model on the residuals (Gradient Boosting) or you need to weight based on predictions (AdaBoost)
- Can only parallelize in the tree-building step (this is where additional cores come in)

**For Bagging/Random Forests/Extra Trees**, each iteration does not rely on a previous iteration

- We just take bootstrap samples or use different number of predictors within each tree.