



BOOSTING TREES

Paul Speaker

Review of Bagging

- Bagging is an **ensembled** model
 - There are many models built, and the predicted are built off of the average
 - Method can be applied to many different models, but usually for decision trees
- The models are different from bagging because the training data is resampled each time
 - Resampled with replacement (like bootstrapping) where individual datapoints can be left out or duplicated
- Because there is a lot of overlap, the trees are correlated
- Random forests are a type of bagging where
 - A random subset of features are considered each time
 - Each tree is forced to be small

Boosting with Trees

- **Boosting** is another ensembling approach
- Rather than forming models independently, each model learns sequentially from the previous one
- Also can be for any model, but most typically for decision trees
- What does it mean to learn from previous models?
 - Build a tree
 - Target for new tree is not original target
 - Instead, new tree is built with use errors of previous trees
 - With different target each time, the trees are going to look very different from each other
 - **Decorrelated** Trees

Boosting with Trees

- **Boosting** is another ensembling approach
- Rather than forming models independently, each model learns sequentially from the previous one
- Also can be for any model, but most typically for decision trees
- What does it mean to learn from previous models?
 - Build a tree
 - Target for new tree is not original target
 - Instead, new tree is built with use errors of previous trees
 - With different target each time, the trees are going to look very different from each other
 - **Decorrelated** Trees

Types of Boosting

- Adaptive Boosting
 - After first model is built, data is reweighted so that misclassified examples have a higher weight
 - New model built off of weighted data
 - Reweighting happens again and build new models
 - Prediction is ensembled off of all trees
- Gradient Boosting
 - After first model is built, residuals are calculated
 - New tree is built off of residuals instead of original target
 - Keep going

Gradient and Extreme Gradient Boosting

- Extreme gradient boosting allows for better control for overfitting vs. regular gradient boosting
- There are 3 main hyperparameters for gradient boosting
 - Number of trees
 - Gives the number of iterations
 - 500-1000 is typical for gradient boosting
 - Learning rate (also called regularization parameter)
 - The weight for implementing misclassification values
 - Typical value is ~ 0.1
 - Number of splits/max depth of tree
 - How large the trees are
 - Trees are typically very small (rarely $\text{maxdepth} > 2$)
 - Called “weak learners” because trees are small

Packages for Doing Boosting

- Adaptive boosting
 - Adaboost in JOUSBoost
- Gradient Boosting
 - gbm in gbm library
- Extreme Gradient Boosting
 - xgboost in xgboost library
 - The one used the most in data science
 - Probably the single most used data science classification algorithm
- xgboost exists in Python also
- Given the importance of xgboost in data science, we will continue its use next time!

Packages for Doing Boosting

- Adaptive boosting
 - Adaboost in JOUSBoost
- Gradient Boosting
 - gbm in gbm library
- Extreme Gradient Boosting
 - xgboost in xgboost library
 - The one used the most in data science
 - Probably the single most used data science classification algorithm
- xgboost exists in Python also
- Given the importance of xgboost in data science, we will continue its use next time!

xgboost Syntax

- Only numerical features (categorical are experimental for now)
- Separate x and y
- x is a matrix
- Hyperparameters
 - nrounds—number of trees to iterate
 - eta –learning rate
 - max_depth—maximum tree size
- xgb_importance—interpret the tree outputs