MORE XGBOOST

Paul Speaker

Review of xgboost

- xgboost is the single most important package for data science predictive modeling
- 3 main hyperparameters
  - Number of trees
    - Gives the number of iterations
    - 500-1000 is typical for xgboost
  - 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 maxdepth > 2)
    - Called "weak learners" because trees are small

Getting the Best Hyperparameters

- Need to search systematically for best values
  - **Hyperparameter tuning—**the search for the best hyperparameters

- There are several methods for hyperparameter tuning, which range from manual to optimized
  - Trial and error (tried this one already!)
  - Grid search
    - create a set of hyperparameter combinations, try them all
  - Optimized search capabilities
    - Define limits for hyperparameters
    - Perform a targeted search

Getting the Best Hyperparameters

- Each method has advantages and disadvantages
- Trial and Error
  - Advantages: Easy to try, can learn a lot by manually trying
  - Disadvantages: Too many combinations to try efficiently
- Grid search
  - Advantage: Covers all possibilities
  - Disadvantage: computational cost—may be too many possibilities to run
    - Scales with product of different hyperparameter values
- Optimized search
  - Advantage: Can be more efficient than full grid search
  - Disadvantage: often will converge to local minimum error value, not global

Other Considerations for Search

- Rather than train-test split, doing cross-validation is most common method for hyperparameter search
- Most data scientists use uniform grid when doing grid search
  - Equally-spaced hyperparameter values
  - For integer-valued hyperparameters, this is only option
  - For hyperparameters which span different scales, a log grid is faster
    - Example:
      - eta (learning rate)—typically scales from 0.01 to 0.3
      - Difference between 0.01 and 0.02 is significant, but difference between 0.29 and 0.3 is not
    - A log-scaled grid allows fewer point to cover same span
- Combinations of approaches
  - Coarse random search, fine grid search
    - Try ~10 random points in hyperparameter space, do finer grid search where random is minimum

Hyperparameter search with caret

- The caret package in R allows for easy setup for grid search for hyperparameters
- This is done through the trControl setting
- Overall process:
  - Define a dataframe with all hyperparameters
    - "All" is more than the 3 discussed previously. The others can simply be fixed values
  - Pass the dataframe to a tuning control object
  - Set method (to cv for cross-validation)
  - Use caret to train the model with the above tuning controls
- Will output accuracies for all combinations
- Plots of accuracies vs. hyperparameter values are very interesting