ENSEMBLING MODELS
BAGGING
RANDOM FORESTS

Paul Speaker

Ensembling Models

- Suppose you have created 2 or more different models off data for the same targets
- Often, “combining” model results can be better than any of the individual models
  - **Ensembled model**
- Usually, ensembled models are from different sources
  - Different model types
  - Different data sources (re-sampling)
  - Usually does not help to have models of same model from same data sources (even if different X’s from data are used)

Ensembling Models

- How to combine different model results?
- For numeric target, simply average
- For classification, Raw material for combining comes from the prediction probabilities
- Suppose we have 3 different classification models. We can combine by
  - Simple average of prediction probabilities → use average in predictions
  - Weighted average of prediction probabilities
    - Can do “best fit” to find weights
    - Weights add to 1
      - For simple average, weights are equal
- Occasionally will see majority vote on classification—**Do not recommend**

Bagging Models

- Although decision tree models struggle by themselves, they are particularly useful through ensembling methods
  - Main methods for ensembling trees are
    - Bagging
    - Boosting
  - We will cover bagging today
- Basic idea for bagging is straightforward
  - We split data into train/test or CV splits as usual
  - We re-sample data in train set
    - Just like with bootstrapping, so with replacement
    - Build a decision tree on resampled data
    - Repeat many times
    - Run models on test dataset, take average of probabilities (called out-of-bag)

Bagging Models

- Note that the general process for bagging works regardless of the type of model
  - For many models it does not help
  - However, it is particularly effective with decision trees
- Just like with ordinary trees, each individual tree in the bag can be pruned
  - My experience: pruning is not necessary for bagged trees (and often reduces accuracy)
  - Errors for overfit trees tend to fall out in the large averages
- Doing a lot of trees for large datasets is computationally expensive
  - Even a smaller number of trees can work well

Random Forests

- Random forests are a modified form of bagged tree models
- Basic approach for bagging is followed for random forest
  - Train dataset is resampled
  - New decision tree is built from train dataset
  - Results from trees are averaged for ensembled prediction
- Two modifications
  - Fixed depth of trees (number of times a node can be split)
  - Each time, a random subset of features are considered
    - Typically, less than ½ of possible features are considered each time
      - Square root often used
- Rationale for this approach
  - Prevents case where a very small number of features dominate model
  - More diverse set of trees (***decorrelates trees***)
  - Smaller number of splits → faster than straight bagging

Random Forests

- Random forests in R can be done with the randomForest library
- Can specify
  - Number of trees (ntree =)
  - Number of features to try each iteration (mtry =)
  - Maximum number of nodes in each tree (maxnodes =)
  - Variable importance tracking (importance = TRUE/FALSE)
  - Resampling with/out replacement (Replace = TRUE/FALSE)
- Variable importance tracking allows to measure which variables have the largest impact on the model
  - Important key to interpreting random forests
- Regular bagging
  - Number of trees = number of resampling
  - Number of features each time = total number of features
  - No maxnodes
  - Replace = TRUE