STAT430: Machine Learning for Financial Data

LarryHua.com/teaching

Spring 2019

Feature Importance

Mean Decrease Impurity (MDI)

- Fast, explanatory-importance (in-sample, IS) method specific to tree-based classifiers
- At each split, calculate how much impurity (Gini/deviance) is decreased, and attribute it to the corresponding feature
- Therefore, for each tree the overall impurity decrease associated with each feature can be calculated.
- Average across all trees and rank the features accordingly

Mean Decrease Impurity (MDI) - notes

- Masking effects: some features may not be used for split
  - Give more chance to less important features, say mtry=1 in randomForest in R
  - When average across all trees, do not count those trees where the feature is not randomly selected for splits
- In-sample performance: every feature will have some importance, even if they have no predictive power
- MDI cannot be generalized to other non-tree based classifiers
- Does not address substitution effects / multi-colinearity

Mean Decrease Accuracy (MDA)

- Slow, predictive-importance (out-of-sample, OOS) method
- Typical steps:
  - Fit a classifier
  - Derive OOS performance (e.g., accuracy, etc.)
  - Permutate each column of the features matrix, one column at a time, to calculate the reduction of performance

Mean Decrease Accuracy (MDA) - notes

- Applicable beyond tree-based classifiers
- Flexible in performance scores, such as F1 rather than accuracy
- When cardinal scores are not available, use ordinal scores.
- Does not address substitution effects / multi-colinearity
- It is possible that all features are unimportant (based on OOS CV)

Single Feature Importance (SFI)

- Predictive-importance (out-of-sample) method
- Compute OOS performance of each feature separately
- Some notes:
    - Applicable beyond tree-based classifiers
    - Flexible in performance scores
    - No substitution effects / multi-colinearity
    - It is possible that all features are unimportant (based on OOS CV)
    - Omitting interaction effects: a classifier with two features can be better than the bagging of two single-feature classifiers
- Try R

Orthogonal Features

- Before applying MDI and MDA, orthogonalize features
- Use principal components analysis (PCA) to reduce linear substitution effects
  - Let $\{X_{t,n}\}, t = 1, \ldots, T; n = 1, \ldots, N$ be stationary features
  - Calculate $Z_{t,n} = (X_{t,n} - \mu_n)/\sigma_n$
  - Form diagonal matrix Λ with the main entries being $Z'Z$'s eigenvalues sorted from large to small
  - Let W be an orthonormal matrix with columns being the eigenvectors such that $Z'ZW = W\Lambda$
  - Use $P = ZW$ as the orthogonal features.

Orthogonal Features - benefits

- benefit of standardizing
  - centering to make the first principal component point to the main direction of the observations
  - re-scaling the data to focus on explaining correlations rather than variances
- benefits of orthogonal features
  - dimension reduction by dropping (converted) features with smaller eigenvalues
  - support feature importance analysis by MDI, MDA, or SFI
    - See FIGURE 8.1 of AFML

Implementation of feature importance

- randomForest() in R has already included MDI and MDA, and SFI can be done by including only one feature at a time
- However, randomForest() does not use sequential bootstrap
- Hack randomForest()
  - randomForestFML
- Try R
- Back to Course Scheduler