BUS 41204 Review Session 5

Variable Selection

Siying Cao
siyingc@uchicago.edu

2/4/2017

Plan

- Why variable selection
- Two main approaches
  - wrapper methods
  - filter methods
- Example using **recursive feature elimination** method

Why Variable Selection

- Determining which predictors to include in a model is crucial
  - model is more interpretable with fewer predictors
  - some models negatively affected by non-informative/redundant predictors
  - some models such as regression trees and LASSO are naturally resistant to non-informative predictors

Main Approaches For Reducing the Number of Predictors

- Wrapper methods
  - Evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance
  - Pros: Selection criterion directed related to model effectiveness
  - Cons: computationally intensive; risk of over-fitting
- Filter methods
  - Evaluate the relevance of predictors outside of the predictive models and subsequently model only the predictors that pass some threshold criterion
  - Correlation matrix is helpful for detecting relevant predictors
  - Pros: computationally efficient
  - Cons: redundant predictors may be selected; interactions hard to quantify

Wrapper Methods

- Classical forward selection for linear regression
  - Problems
    1. Greedy search algorithm, hence does not reevaluate past solutions
    2. Repeated hypothesis tests invalidates many statistical properties since the same data are being evaluated multiple times
    3. Maximizing statistical significant $\neq$ predictive accuracy
- Stepwise selection
- Backward selection via recursive feature elimination (RFE)

Naive Recursive Feature Elimination

- Algorithm

Tune/train the model on the training set using all P predictors
Calculate model performance
Calculate variable importance or rankings
for *each subset size* S_i, $i = 1 \ldots S$ **do**
    Keep the S_i most important variables
    [Optional] Pre-process the data
    Tune/train the model on the training set using S_i predictors
    Calculate model performance
    [Optional] Recalculate the rankings for each predictor
end
Calculate the performance profile over the S_i
Determine the appropriate number of predictors (i.e. the S_i associated with the best performance)
Fit the final model based on the optimal S_i

Selection Bias

- The same idea as over-fitting. Feature selection is part of the model selection process!!
- The risk of over-fitting increases if
  1. The dataset is small
  2. The number of predictors is large
  3. The predictive model is powerful, which is more likely to overfit the data
  4. No independent test set is available
- Solution
  - For large data set, separate data sets for selecting features (training set), tuning models (validation set), and validating the final model (test set)
  - For small data set, use **cross-validation**

Recursive Feature Elimination with Cross Validation

- Algorithm

for *each resampling iteration* **do**
    Partition data into training and test/hold-back set via resampling
    Tune/train the model on the training set using all P predictors
    Calculate model performance
    Calculate variable importance or rankings
    for *Each subset size* S_i, $i = 1 \ldots S$ **do**
        Keep the S_i most important variables
        [Optional] Pre-process the data
        Tune/train the model on the training set using S_i predictors
        Calculate model performance using the held-back samples
        [Optional] Recalculate the rankings for each predictor
    end
end
Calculate the performance profile over the S_i using the held-back samples
Determine the appropriate number of predictors
Determine the final ranks of each predictor
Fit the final model based on the optimal S_i using the original training set

Hands-on Implementation

See the companion R file

- TA5_code_my.R: coding from scratch (to demonstrate what's inside the blackbox)
- TA5_code_caret.R: using caret package

References

- RFE implmentation in R: caret package

https://topepo.github.io/caret/recursive-feature-elimination.html

- RFE implementation in Python: sklearn.feature_selection.RFECV module

http://scikit-learn.org/stable/modules/feature_selection.html

- Book on feature selection

Kuhn, Max, and Kjell Johnson. Applied predictive modeling. Vol. 26. New York: Springer, 2013