5. Filtering & Variable Selection

Statistical Modelling & Machine Learning

Jaejik Kim

Department of Statistics
Sungkyunkwan University

STA3036

Why Filtering / Selecting Variables?

- Prediction: Irrelevant input variables make data pattern unclear (curse of dimensionality) and overfitting occurs.
- Interpretation: Removing irrelevant variables $\Rightarrow$ Removing unnecessary complexity of model $\Rightarrow$ Interpretability $\uparrow$.
- Filtering variables:
  - Evaluating input variables before training models.
  - For very high-dimensional data, penalization methods such as lasso might not work correctly.
  - For a large p, filtering is required (both supervised and unsupervised learnings).
- Variable selection: Selection of variables in the final prediction model (supervised learning).

Variable Importance Measures

- In large p situations, filtering input variables might be required for effective predictive modelling.
- Some filtering methods are based on measures for the importance of individual variables.
- Variable importance $\Rightarrow$ Ranking of variables.
- Remind the variable importance in random forests (permutation idea).

Variable Importance in Regression

Y: Continuous; X: Continuous

- Pearson correlation coefficient: Linear association.
- Spearman's rank correlation coefficient: Nearly linear or curvelinear relationships.

$$r_S = \frac{Cov(X_R, Y_R)}{S_{X_R} S_{Y_R}},$$

  - X_R and Y_R: Rank variables converted from X and Y, respectively.
  - S_{X_R} and S_{Y_R}: Sample standard deviation of X_R and Y_R, respectively.
  - Pearson correlation coefficient between rank variables.

Variable Importance in Regression

Y: Continuous; X: Continuous

- Pseudo R^2: Nonlinear relations.

$$Pseudo\ R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}.$$

  - Fit a nonparametric smoother (e.g., local linear regression for (X, Y)) to data, and then compute Pseudo R^2.

Variable Importance in Regression

Y: Continuous; X: Continuous

- Maximal information coefficient (MIC): Linear and nonlienar relationships (most functional types).

$$MIC(X, Y) = \frac{\hat{I}(X, Y)}{\log_2\{\min(m_X, m_Y)\}},$$

  - Mutual information: $\hat{I}(X, Y) = \sum_{\tilde{x},\tilde{y}} \hat{p}(\tilde{x}, \tilde{y}) \log_2 \frac{\hat{p}(\tilde{x},\tilde{y})}{\hat{p}(\tilde{x})\hat{p}(\tilde{y})}$, where $\hat{p}(\tilde{x}, \tilde{y})$ is the fraction of data points falling into bin $(\tilde{x}, \tilde{y})$, and m_X and m_Y are the number of bins on X and Y axes, respectively.
  - It has a value between 0 and 1.
  - $MIC(X, Y) = 0$: Independence of X and Y.
  - $MIC(X, Y) = 1$: Completely noiseless relationship.

MIC

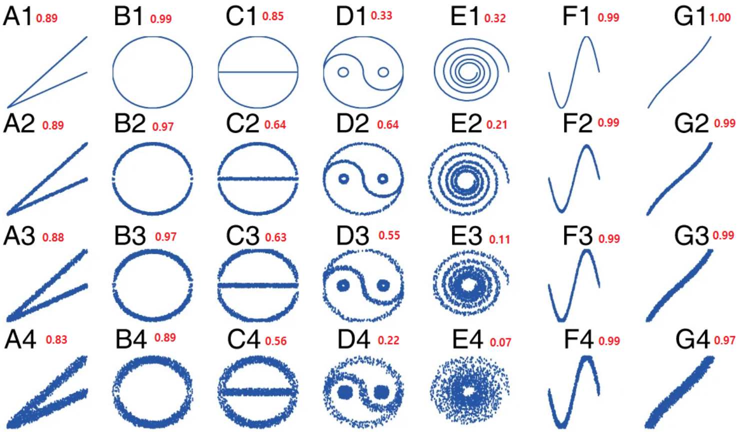

Figure: MIC values (Zhang et al., 2014; Scientific Reports, volume 4, Article number: 6662)

Variable Importance in Regression

Y: Continuous; X: Categorical

- Binary input variables:
  - t-statistic (or p-value) from t-test (normal assumption).
  - Wilcoxon rank sum test statistic (no distributional assumption).
- Input variables with three or more categories:
  - F-statistic from one-way ANOVA (normal assumption).
  - Kruskal-Wallis one-way ANOVA (no distributional assumption).

Variable Importance in Classification

Y: Categorical; X: Continuous & Categorical

- Relief algorithm:
  - It works for a binary Y, but it can be extended into Y with multi-class by applying the algorithm to each class.
  - All continuous inputs should be transformed into [0,1] scale.
  - Categorical inputs should be encoded by 0 or 1.
  - At each iteration, it randomly select a training obs (say $\boldsymbol{x}_i$; $p \times 1$ vector).
  - Find the nearest training obs. in $Y = 0$ and $Y = 1$ classes to $\boldsymbol{x}_i$.
  - Let $\boldsymbol{x}_H$ (Hit) be the nearest training obs. in the same class as the class of $\boldsymbol{x}_i$ and $\boldsymbol{x}_M$ (Miss) be the near training obs. in the other class.
  - It uses the difference of $(\boldsymbol{x}_i, \boldsymbol{x}_H)$ and the difference of $(\boldsymbol{x}_i, \boldsymbol{x}_M)$.
  - Difference of $(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = [(x_{i1} - x_{i'1})^2, \ldots, (x_{ip} - x_{i'p})^2]^\top.$$

Variable Importance in Classification

Relief algorithm:

1. Initialize the $p \times 1$ score vector $\boldsymbol{S} = \mathbf{0}$.
2. For $k = 1, \ldots, K$,
   - 2-1. Randomly select a training obs. $\boldsymbol{x}_i$ from the training set.
   - 2-2. Find a hit $\boldsymbol{x}_H$ and miss $\boldsymbol{x}_M$ closest to $\boldsymbol{x}_i$
   - 2-3. Update $\boldsymbol{S}$ by

$$\boldsymbol{S} \leftarrow \boldsymbol{S} - d(\boldsymbol{x}_i, \boldsymbol{x}_H) + d(\boldsymbol{x}_i, \boldsymbol{x}_M).$$

$\Rightarrow$ Output: $\boldsymbol{S} = (S_1, \ldots, S_p)^\top$, where S_j is the score of X_j variable.

Variable Importance in Classification

- If the hit is far from $\boldsymbol{x}_i$ in X_j space, then S_j decreases. But, if the miss is far away, S_j increases.
- S_j measures the separability of $Y = 0$ and 1 classes in terms of X_j.
- If Y has K classes and $K > 2$, run the Relief algorithm for the kth class and the other $K - 1$ classes, and then sum all S_j values over K runs of Relief algorithm.
- ReliefF algorithm:
  - Every training obs. becomes $\boldsymbol{x}_i$ (n iterations).
  - At every iteration, it finds k nearest hits and misses.
  - For multi-class, it finds k nearest misses from each class, and then take the average of their contributions for updating $\boldsymbol{S}$.

Limitation of Variable Importance

- Variable importance evaluates each input variable without considering the others.
- Problem 1:
  - If two inputs are highly correlated with Y and with each other, then they will be identified as important variables.
  - In that case, some predictive models will be negatively impacted by this redundant information (e.g., multicollinearity).
- Problem 2:
  - It will miss groups of input variables that together have a strong relationship with Y.
  - No marginal relationship, but strong joint relationship.
- Problem 3: No threshold for variable importance.

Variable Selection

- For better prediction and interpretation, it is important to remove redundant variables and non-informative variables.
  $\Rightarrow$ Variable selection (feature selection).
- Models robust to non-informative variables: Tree model, random forests (more trees are required).
- However, most models are negatively impacted by non-informative input variables.
- Variable selection: Methods to find the optimal subset of input variables that maximizes model performance.
  - Regularization methods: Lasso, Elastic net, SCAD, MCP, etc.
  - Subset selection approaches: Best subset selection, Forward stepwise selection, Backward stepwise selection, hybrid approach.

Subset selection: Simulated Annealing

- Subset selection can be considered as an optimization problem
  $\Rightarrow$ Finding the optimal subset minimizing test error.
- Simulated annealing: A probabilistic technique for approximating the global optimum in a large search space of an objective function.
  - Heuristic algorithm and finite discrete search space.
  - It picks a random move instead of picking the best move.
  - If the randomly selected move improves the optimization, then it is always accepted.
  - Otherwise, it accepts the move with a probability that decreases exponentially with the 'badness' of the move.

Subset selection: Simulated Annealing

Algorithm:

1. Generate an initial random subset of $X_1, \ldots, X_p$.
2. For $k = 1, \ldots, K$,
   - 2-1. Randomly perturb the current best subset $\mathcal{M}_{best} \Rightarrow \mathcal{M}$ (randomly perturbed subset).
   - 2-2. Train the model with the current subset.
   - 2-3. Compute the performance measure E_k (e.g., AIC, BIC, or LOOCV, etc.).
   - 2-4. If $E_k < E_{best}$, accept the current subset $\mathcal{M}$ and set $E_{best} = E_k$ and $\mathcal{M}_{best} = \mathcal{M}$.

Subset selection: Simulated Annealing

- Algorithm (Continued):

  2-5. Otherwise, Compute the probability of accepting the current subset $\mathcal{M}$ by $p_k = \exp\{(E_{best} - E_k)/T\}$, where T changes over iterations. At higher values of T, uphill moves are more likely to occur. In a typical simulated annealing, T starts high and is gradually decreased according to an 'annealing schedule'.

    2-5-1. Generate a uniform (0,1) random number U.

    2-5-2. If $p_k \geq U$, accept the current subset $\mathcal{M}$ and set $E_{best} = E_k$ and $\mathcal{M}_{best} = \mathcal{M}$.

    2-5-3. Otherwise, keep the current $\mathcal{M}_{best}$

3. Find the subset with the smallest E_k across all iterations.

Selection Bias

- Selection bias: Bias introduced by selecting variables.
- Situations that selection bias increases:
  - Small size of data.
  - The number of predictors is large. For large p situations, the prob. of non-informative inputs being falsely declared to be important increases.
  - The complex models are more likely to overfit the data.
  - No independent test set is available.
- For proper evaluation of predictive models with filtering or variable selection, CV or bootstrap procedure should include such steps.

Variable Selection When $p \gg n$

- When $p \gg n$, selection bias is very severe.
- When $p \gg n$, linear models (simple models) have better prediction than nonlinear models (complex models).
- For variable selection in linear models, regularization methods such as lasso, SCAD, or MCP, etc. can be considered.
- However, when $p \gg n$, they might not perform well due to statistical accuracy and algorithmic stability.
- In usual, a single regularization method identifies many irrelevant variables as important variables.

ISIS (Iterative Sure Independence Screening)

- ISIS (Fan et al., 2011): Extension of SIS (Sure Independence Screening).
- SIS and ISIS works for all generalized linear models (e.g., linear regression, logistic regression, Cox regression, etc.).
- SIS consists of two steps:
  - By the size of the marginal MLE (MMLE) $|\hat{\beta}_j^M|$, select d input variables. Typical choice of $d = \llcorner n/\log n \lrcorner$.
  - Apply a regularization method to the model with selected d input variables.
- SIS fails in the following situations:
  - Inputs are marginally unrelated, but jointly related with $Y \Rightarrow$ They should be included in the final model.
  - Inputs are jointly uncorrelated with Y, but have higher marginal correlation than some important inputs. $\Rightarrow$ They should be excluded from the final model.

ISIS

Algorithm: Vanilla ISIS

1. Set initial screening model size d, the type of penalty $p_\lambda(\cdot)$, and the maximum iteration number l_{max}.

2. For $j = 1, \ldots, p$, compute the MMLE $\hat{\beta}_j^M$ from the GLM for Y and X_j. Then, select the $k_1 = \llcorner 2d/3 \lrcorner$ top ranked inputs to form the index set $\hat{\mathcal{A}}_1$ by the size of $\hat{\beta}_j^M$.

3. Apply the penalized ML estimation on the set $\hat{\mathcal{A}}_1$ to obtain a subset of indices $\hat{\mathcal{M}}_1$.

Algorithm: Vanilla ISIS (Continued)

4. Set $l = 2$ and iterate until $|\hat{\mathcal{M}}_l| = d$, $\hat{\mathcal{M}}_l = \hat{\mathcal{M}}_{l-r}$, or $l = l_{max}$:
   4-1. For every $j \in \hat{\mathcal{M}}_{l-1}^C$, compute the conditional marginal MLE (CMMLE) $\hat{\beta}_j^{CM}$ from the GLM for Y and X's with indices $\{\hat{\mathcal{M}}_{l-1}, j\}$.
   4-2. Select the $k_l = d - |\hat{\mathcal{M}}_{l-1}|$ top ranked inputs to form the index set $\hat{\mathcal{A}}_l$ by the size of $\hat{\beta}_j^{CM}$, $j \in \hat{\mathcal{M}}_{l-1}^C$.
   4-3. Apply the penalized ML estimation on $\hat{\mathcal{M}}_{l-1} \cup \hat{\mathcal{A}}_l$ to obtain a new index set $\hat{\mathcal{M}}_l$.

$\Rightarrow$ Output: Final index set $\hat{\mathcal{M}}_l$.

ISIS

Algorithm: Permutation-based ISIS

1. Set initial screening model size d, the type of penalty $p_\lambda(\cdot)$, quantile q, and the maximum iteration number l_{max}.

2. For $j = 1, \ldots, p$, compute the MMLE $\hat{\beta}_j^M$ from the GLM for Y and X_j.

3. Generate a randomly permuted dataset on $(\boldsymbol{x}_i, y_i)$ and obtain the MMLE $\hat{\beta}_j^{M*}$ from the permuted data.

4. Let w_q be the qth quantile of $\{|\hat{\beta}_j^{M*}|;\ j = 1, \ldots, p\}$. Then, Form the index set $\hat{\mathcal{M}}_1 = \{1 \leq j \leq p;\ |\hat{\beta}_j^M| \geq w_q\}$.

5. Apply the penalized ML estimation on the set $\hat{\mathcal{A}}_1$ to obtain a subset of indices $\hat{\mathcal{M}}_1$.

ISIS

Algorithm: Permutation-based ISIS (Continued)

6. Set $l = 2$ and iterate until $|\hat{\mathcal{M}}_l| = d$, $\hat{\mathcal{M}}_l = \hat{\mathcal{M}}_{l-r}$, or $l = l_{max}$:
   - 6-1. For every $j \in \hat{\mathcal{M}}_{l-1}^C$, compute the CMMLE $\hat{\beta}_j^{CM}$ from the GLM for Y and X's with indices $\{\hat{\mathcal{M}}_{l-1}, j\}$.
   - 6-2. Generate a randomly permuted dataset on only the variables in $\hat{\mathcal{M}}_{l-1}^C$ and obtain the CMMLE $\hat{\beta}_j^{CM*}$ from the permuted dataset.
   - 6-3. Let w_q be the qth quantile of $\{|\hat{\beta}_j^{CM*}|;\ j \in \hat{\mathcal{M}}_{l-1}^C\}$. Then, Form the index set $\hat{\mathcal{A}}_l = \{j \in \hat{\mathcal{M}}_{l-1}^C;\ |\hat{\beta}_j^{CM}| \geq w_q\}$.
   - 6-4. Apply the penalized ML estimation on $\hat{\mathcal{M}}_{l-1} \cup \hat{\mathcal{A}}_l$ to obtain a new index set $\hat{\mathcal{M}}_l$.

$\Rightarrow$ Output: Final index set $\hat{\mathcal{M}}_l$.

Implementation of ISIS

- Other choices of d: $\llcorner n/(2\log n) \lrcorner$ or $\llcorner n/(4\log n) \lrcorner$.
- Variable selection when $p < n$:
  - ISIS can be used.
  - Set $d = p$.
  - Instead of the penalization methods, AIC, BIC, or C_p can be used in ISIS.
- To reduce the number of false positive, set the quantile parameter$q = 1$ (the maximum size of coefficient form permuted data).