Chapter 9: Additive models, trees and related methods

9.1: Generalized additive models (GAMs)

- GAMs can be expressed as a function:

$$\boldsymbol{Y} = g[\mu(\boldsymbol{X})] = \alpha + f_1(\boldsymbol{X}_1) + f_2(\boldsymbol{X}_2) + \ldots + f_p(\boldsymbol{X}_p)$$

Among which,

$\boldsymbol{Y}$ (response) ~ exponential family (distribution)

$g(\mu)$ is the link function (identity, log, logit, etc.)

$\boldsymbol{X}_j$ is the predictor

f_j is non-parametric smooth function

9.1.1: Fitting additive models

- While GLMs (generalized linear models) usually use the least squares method (for linear regression) or maximum likelihood method (for logistic regression or others) to estimate the parameter β_j in $\boldsymbol{Y} = g[\mu(\boldsymbol{X})] = \beta_0 + \beta_1\boldsymbol{X}_1 + \beta_2\boldsymbol{X}_2 + \ldots + \beta_p\boldsymbol{X}_p$, or say, "fitting the model", GAMs also use some algorithms with similar considerations to estimate f_j, among which an algorithm called *backfitting* is introduced.
- For linear additive models, *backfitting* algorithm is based on the minimization of PRSS (penalized RSS). RSS seems not so reliable when the numbers of parameters are large or columns of $\boldsymbol{X}$ are correlated with other, hence a penalty with a regularization parameter λ is introduced. The detailed derivations are too hard for me to understand… (why the conventional assumption is that $\sum_1^N f_j(x_{ij}) = 0$? Why then $\alpha = \text{ave}(y_i)$?...)
- For logistic additive models or others, *backfitting* algorithm is based on a penalized log-likelihood, which can be achieved by IRLS (iteratively reweighted least squares) algorithm.

9.1.2: Example: additive logistic regression

- An example of using a local scoring algorithm (a backfitting algorithm within a Newton-Raphson procedure) for additive logistic regression. In medical settings, logistic model for binary outcome can be used for (1) understanding the roles of prognostic factors and (2) classifying individuals in risk screening.
- An example of using logistic additive model to predict spam mails. In the fitting stage, a cubic smoothing spline is used for each predictor $\boldsymbol{S}_j(\lambda)$, which I do not understand… In the interpretation part, it seems that every predictor can be expressed as a composition of a linear component (weighted least squares linear fit) and a nonlinear component (residual). Since the nonlinearity usually happens near 0, it seems we can change the model at 0 count into a linear logistic model, instead of a GAM, in order to decrease error rate. (not quite understood…)
- Commonly the tolerable error has a direction. In the spam email example, it is more tolerable if we mistakenly put spam mails into normal mails, while we require no normal mail to be classified into spam mail, then L_{10} should be smaller than L_{01}. Then I get lost again at the estimated Bayes rule in predicting the error rate…

- After fitting, the effect of interaction on fitting should be checked.

9.1.3: Summary

- GAMs are flexible while interpretable.
- Limitation of GAMs: not so feasible when the number of predictors is large. Some algorithm other than *backfitting* may be better in dealing with large data.

9.2: Tree-based methos

9.2.1: Background

- CART (classification and regression tree): when the output Y is categorical it is a classification tree, and when Y is continuous it is a regression tree.
- The commonly used partition in CART is recursive binary splitting, as expressed as: (2 inputs→2-dimensional)

  $$\hat{f}(\boldsymbol{X}) = \sum_{m=1}^{M} c_m I\{(\boldsymbol{X}_1, \boldsymbol{X}_2) \in = \boldsymbol{R}_m\}$$

  And the top right panel in Figure 9.2.
- The partition can also be expressed as a top down tree, which is easy to understand (interpretable).

9.2.2: Regression tree (output is continuous)

- (1) Growing a regression tree (estimation of parameters, "fitting"): based on a criterion similar to least squares method.

  * deciding the constant output c_m: minimization of the sum of squares $\sum(y_i - f(x_i))^2 \Rightarrow$ the constant output c_m for region $\boldsymbol{R}_m$ can be estimated as the average of y_i in $\boldsymbol{R}_m$.

  * deciding the splitting point s for splitting variable j: similar considerations
- (2) Pruning a regression tree (decide the size of the tree)

  * How large the tree should be? → Pruning: too large → information seems to be precise but observation numbers in each node become less and less → hardly able to show the characteristics of the data/predict → overfitting → pruning is necessary

  * Growing a big tree T_0 → post-pruning: evaluation of the error rate and complexity after pruning: *cost-complexity pruning*, which can be expressed as

  $$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

  Which can be regarded as a loss function? (the loss after pruning a small tree T_α = increase of the error rate + complexity)

  * deciding α by x-fold cross-validation (the α that can minimize error rate) → find the only smallest subtree that T_α that can minimize $C_\alpha(T)$ → completing pruning (not so sure…)

9.2.3: Classification tree (output is categorical)

- Both growing and pruning a classification tree are based on **node impurity**, which is a measure of the homogeneity of the labels at the node (indicating the prediction ability of the classification tree)
- 3 measures for node impurity

  (1) Misclassification error: the proportion of the training samples that are not classified into the majority class

  (2) Gini index: when p or 1-p is close to 0, Gini index is very small → pure

  (3) Cross-entropy or deviance: similar to Gini index

  (1) is usually used in pruning, while both (2) and (3) (they are similar) are more sensitive in differentiate so they are used in growing a tree
- Gini index can also be explained as the training error rate or the variance over a node (not quite understood…)

9.2.4: Other issues

- Categorical predictors: levels of the predictor *q* ↑, information ↑, the chance of having a good choice ↑, however partitions also ↑↑, overfitting might occur. Solution: give them orders? (not so sure…)
- The loss matrix: the tolerable loss may have a direction, which means not the same for 2 directions. The loss matrix can be used in growing the model, such as being incorporated in Gini index for multiclass cases, or as a weight for pretest probability in classifying the majority class $k(m)$. (not quite understood…)
- Missing predictor values: in GLMs, we use imputation to deal with missing data, while the tree-models have their own ways to deal with missing values. (1) Make a new category "missing". (is it also applicable in GLMs, when the variable containing missing values is categorical?) (2) Construction of surrogate predictors (?). The principle for the latter is to find correlation between the missing predictors and the others, if there is correlation then surrogate predictors can be used (similar as imputation of MAR, in which missing values are predicted according the regression of known variables).
- The reason that using binary splits is mainly because if using multiple splits then the observation numbers would decrease too fast → statistical power might decrease fast
- Limitations of tree-based models:

  (1) Instability: due to the hierarchical nature; can be improved by some algorithm, but the inherent instability cannot be removed.

  (2) Lack of smoothness

  (3) Difficulty in capturing additive structure

9.2.5: Spam example

- Expansion of the error rate → sensitivity (ability to find the true one among the real ones) and specificity (ability to find the false one among the true false one) → ROC curves and AUC (c-statistic)

- C-statistic: comparing the performances of different models, however not efficient in comparing the same model but with different numbers of predictors