

9

Additive Models & Trees

19.03.08
임소현

CONTENTS

1

Generalized additive models

2

Tree-Based Methods

3

Example

CONTENTS

1

Generalized additive models

2

Tree-Based Methods

3

Example

1) Generalized additive models

$$E(Y|X_1, X_2, \dots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p).$$

- X_1, X_2, \dots, X_p : predictors
- Y : the outcome
- the f_j 's : unspecified smooth ("*nonparametric*") functions

- we fit each function using a scatterplot smoother (e.g., a cubic smoothing spline or kernel smoother)
- provide an algorithm for simultaneously estimating all p functions


2) Fitting Additive Models

- The additive model has the form

$$Y = \alpha + \sum_{j=1}^p f_j(X_j) + \varepsilon$$

$$\text{PRSS}(\alpha, f_1, f_2, \dots, f_p) = \sum_{i=1}^N \left(y_i - \alpha - \sum_{j=1}^p f_j(x_{ij}) \right)^2 + \sum_{j=1}^p \lambda_j \int f_j''(t_j)^2 dt_j,$$

,where the $\lambda_j \geq 0$ are tuning parameters

- minimizer of (9.7) is an additive cubic spline model
- each of the functions f_j is a cubic spline in the component X_j , with knots at each of the unique values of x_{ij} , $i = 1, \dots, N$
- without further restrictions on the model  the solution is not unique

2) Fitting Additive Models

Algorithm 9.1 *The Backfitting Algorithm for Additive Models.*

1. Initialize: $\hat{\alpha} = \frac{1}{N} \sum_1^N y_i$, $\hat{f}_j \equiv 0, \forall i, j$.
2. Cycle: $j = 1, 2, \dots, p, \dots, 1, 2, \dots, p, \dots$,

$$\hat{f}_j \leftarrow \mathcal{S}_j \left[\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik})\}_1^N \right],$$

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N} \sum_{i=1}^N \hat{f}_j(x_{ij}).$$

until the functions \hat{f}_j change less than a prespecified threshold.

3) Additive Logistic Regression

- The generalized additive logistic model has the form

$$\log \frac{\Pr(Y = 1|X)}{\Pr(Y = 0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p).$$

- The functions f_1, f_2, \dots, f_p are estimated by a backfitting algorithm within a Newton–Raphson procedure
- The additive model fitting in step (2) of Algorithm 9.2 requires a weighted scatterplot smoother.

3) Additive Logistic Regression

Algorithm 9.2 *Local Scoring Algorithm for the Additive Logistic Regression Model.*

1. Compute starting values: $\hat{\alpha} = \log[\bar{y}/(1 - \bar{y})]$, where $\bar{y} = \text{ave}(y_i)$, the sample proportion of ones, and set $\hat{f}_j \equiv 0 \ \forall j$.
2. Define $\hat{\eta}_i = \hat{\alpha} + \sum_j \hat{f}_j(x_{ij})$ and $\hat{p}_i = 1/[1 + \exp(-\hat{\eta}_i)]$.

Iterate:

- (a) Construct the working target variable

$$z_i = \hat{\eta}_i + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}.$$

- (b) Construct weights $w_i = \hat{p}_i(1 - \hat{p}_i)$
- (c) Fit an additive model to the targets z_i with weights w_i , using a weighted backfitting algorithm. This gives new estimates $\hat{\alpha}, \hat{f}_j, \forall j$

3. Continue step 2. until the change in the functions falls below a pre-specified threshold.

CONTENTS

1

Generalized additive models

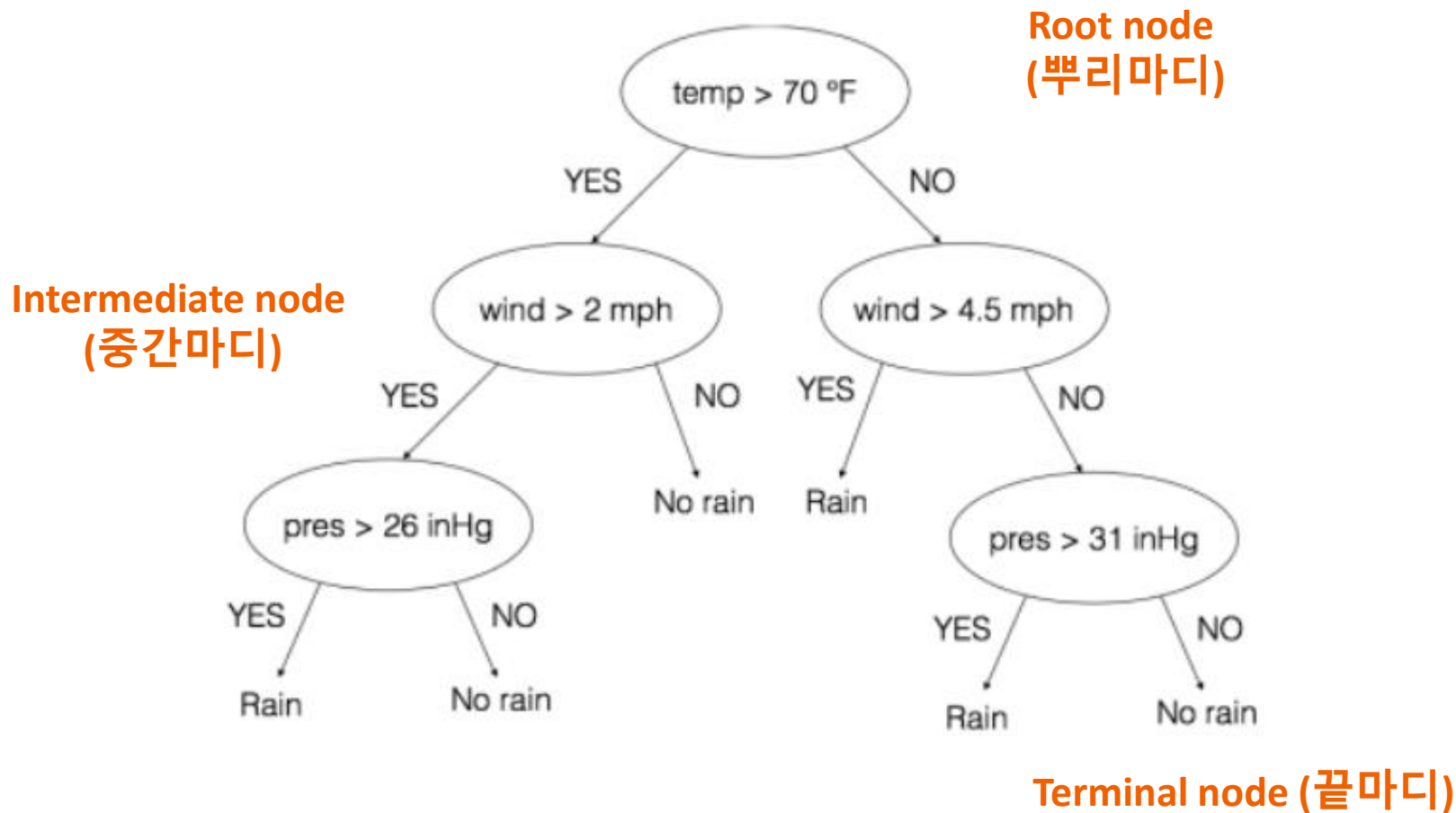
2

Tree-Based Methods

3

Example

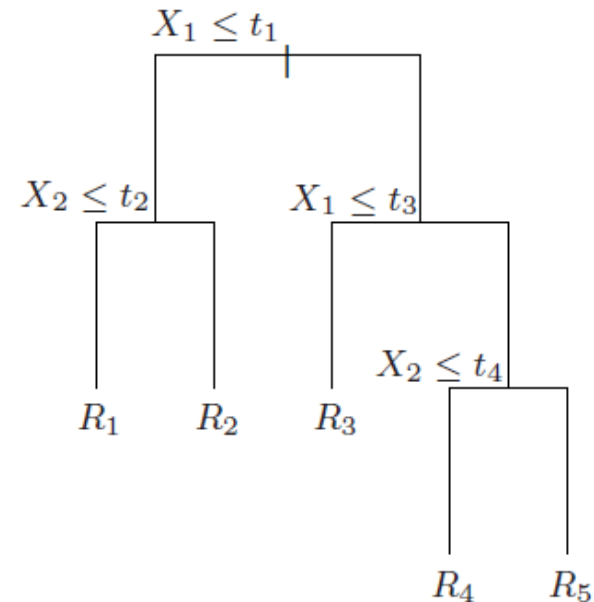
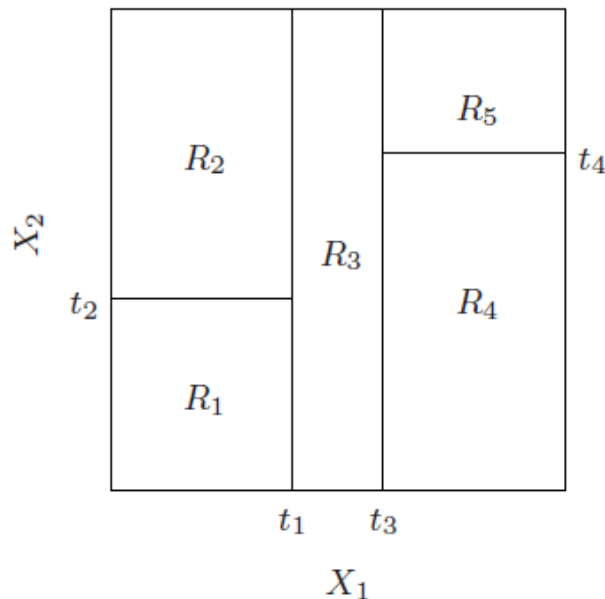
1) Tree-Based Methods



1) Tree-Based Methods

- Tree-based methods partition the feature space into a set of rectangles
- fit a simple model (like a constant) in each one.
- a popular method for tree-based regression and classification called **CART**

Classification And
Regression Tree



2) Regression Trees

- Suppose first that we have a partition into M regions R_1, R_2, \dots, R_M ,
- we model the response as a constant c_m in each region:

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$

- Now finding the best binary partition

$$R_1(j, s) = \{X | X_j \leq s\} \text{ and } R_2(j, s) = \{X | X_j > s\}$$

$$\min_{j, s} \left[\min_{c_1} \sum_{x_i \in R_1(j, s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j, s)} (y_i - c_2)^2 \right]$$



$$\hat{c}_1 = \text{ave}(y_i | x_i \in R_1(j, s)) \text{ and } \hat{c}_2 = \text{ave}(y_i | x_i \in R_2(j, s))$$

2) Regression Trees

- How large should we grow the tree?



a very large tree might overfit the data
a small tree might not capture the important structure

- *cost-complexity pruning*

Find tree which minimizes

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

$$N_m = \#\{x_i \in R_m\},$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i,$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

- Choosing α adaptively by weakest link pruning
- α the tradeoff between tree size and its goodness of fit to the data

3) Classification Trees

- Only change in the criteria to split nodes and pruning the tree
- \hat{p}_{mk} : proportion of class k on node m

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k).$$

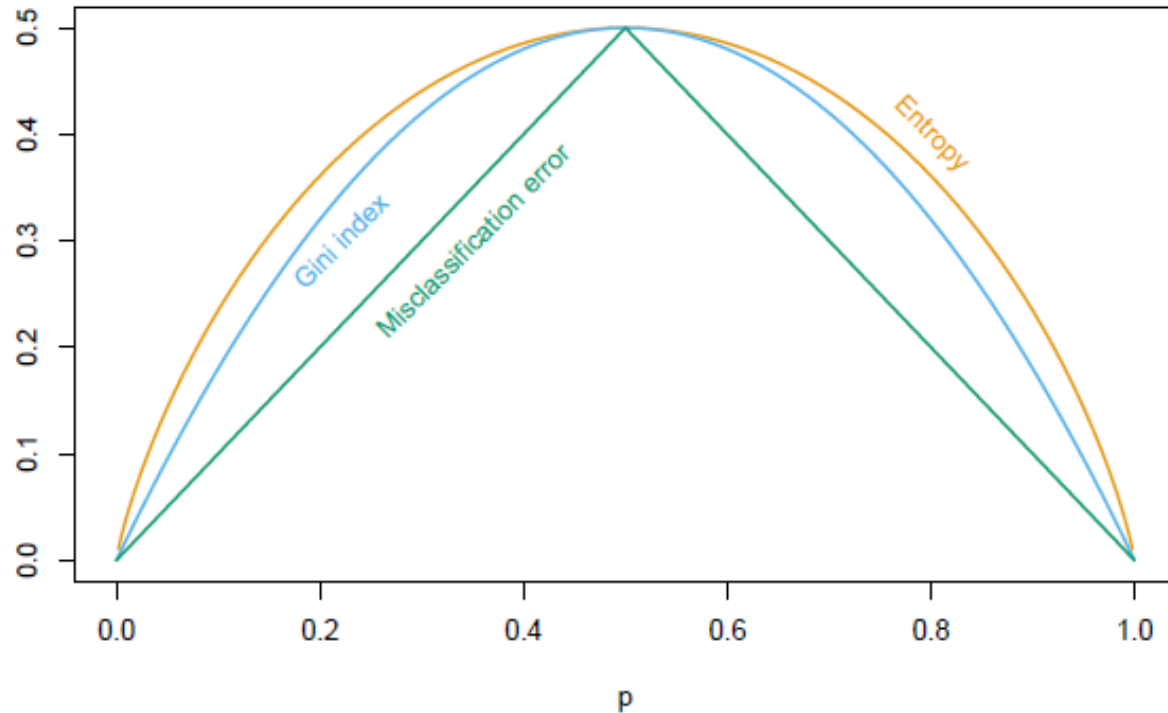
- $k(m)$ be the majority class on node m, i.e. $k(m) = \arg \max_k \hat{p}_{mk}$.
- For each node, partition to minimize

Misclassification error: $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}.$

Gini index: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk}).$

Cross-entropy or deviance: $-\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}.$

3) Classification Trees



- Cross-entropy and Gini index are more sensitive to changes in the node probabilities than the misclassification rate.
- Either cross-entropy and Gini index should be used when growing the tree.

CONTENTS

1

Generalized additive models

2

Tree-Based Methods

3

Example



THANK YOU

