LARS and Boosting

김용대[1]

[1] 서울대학교 통계학과

축소 (Shrinkage, Frieman 2001)

- $F_m = F_{m-1} + T_m$으로 갱신하는 대신, 적당한 작은 상수 $\gamma \in (0, 1)$에 대해서 $F_m = F_{m-1} + \gamma T_m$으로 앙상블 모형을 갱신한다.
- Friedman (2001)은 $\gamma = 0.1$일때 예측력이 좋다는 것을 실증적으로 보였다.
- 이 방법은 축소추정량과 밀접한 관련이 있다.
- 이해하기 쉽지 않은 부분은 축소모수 γ를 작게 할 수록 예측력이 계속 좋아진다는 것이다.
- 놀랍게도, 이러한 이상한 현상을 LARS (Least Angle Regression)으로 설명할 수 있다!

부스팅 알고리즘과 고차원 회귀모형

- 주어진 학습자료 $\mathcal{L}$에 대해서 유한개의 기저예측모형만이 다른 값을 가질 수 있다 (예: 학습자료에서 서로 다른 값을 갖는 의사결정나무는 유한개다)
- 즉, $\mathcal{H} = \{T_1, \ldots, T_q\}$라 할 수 있는데, 여기서 기저예측모형의 개수 q는 매우 크기는 하지만 유한이다.
- 로짓부스팅은 다음의 고차원 로지스틱 회귀모형에서 회귀계수를 추정하는 문제로 이해할 수 있다:

$$\log \frac{\Pr(Y=1|\mathbf{x})}{\Pr(Y=-1|\mathbf{x})} = \beta_0 + \sum_{j=1}^{q} \beta_j T_j(\mathbf{x}).$$

부스팅 알고리즘과 고차원 회귀모형 (계속)

- 즉, 로짓부스팅은 다음의 두 단계로 구성되어 있다:
  1. 입력변수를 의사결정나무를 이용하여 새로운 변수로 변환한다,
  2. 변환된 입력변수들의 선형 로지스틱 회귀모형을 고려하고, 회귀계수를 추정한다.
- 이러한 면에서, 로짓부스팅은 다항회귀와 유사하다고 볼 수 있다:

$$\log \frac{\Pr(Y=1|\mathbf{x})}{\Pr(Y=-1|\mathbf{x})} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{j,k} \beta_{jk} x_j x_k + \sum_{j,k,l} \beta_{jkl} x_j x_k x_l + \cdots .$$

- 의사결정나무를 이용하여 입력변수를 변환하는 방법의 튼 장점은 입력변수의 잡음에 강건하다는 것이다!

LARS 복습

- $\pi = \emptyset$ (i.e. $j = 1$) 인 경우 고려
- $\mathbf{r}(\gamma) = \mathbf{y} - X^{\hat{1}}\gamma$는 잔차
- $\beta_{\hat{1}} > 0$라 하자.
- LARS의 아이디어는 γ를 0부터 증가시키다가, 잔차가 다른 입력변수에 대해서 더 설명이 되면 γ의 증가를 멈춘다.
- 즉, γ를 $\hat{\gamma}$까지 증가시킨다.

$$\hat{\gamma} = \text{argmin}_{\gamma} \left[Corr(\mathbf{r}(\gamma), X^{\hat{1}}) \leq \max_{k \neq hat1} Corr(\mathbf{r}(\gamma), X^{k}) \right].$$

Regularized 부스팅 알고리즘과 고차원 회귀모형

1. 초기화 : $\beta = 0$ 이고 $\epsilon > 0$.
2. $\mathbf{X}$: $n \times q$ 행렬이고 j번째 열이 $(T_j(\mathbf{x}_i), i = 1, \ldots, n)'$ 로 구성
3. 수렴할 때까지 반복한다 :
   1. 잔차계산 : $\mathbf{r} = \mathbf{y} - \mathbf{X}\beta$
   2. 변수선택 : $j = \text{argmin}_k \|\mathbf{r} - X^k \beta_k\|^2$
   3. 회귀계수 갱신 : $\beta_j = \beta_j + \epsilon \ \text{sign}\{Corr(\mathbf{r}, X^j)\}$.

부스팅과 LARS

- Regularized 부스팅 알고리즘에서 ϵ을 0으로 보내면, LARS 알고리즘과 같아진다.
- 즉, Regularized gradient 부스팅알고리즘은 의사결정나무로 변환된 입력변수를 사용하여 로지스틱 모형의 회귀계수를 LARS (and so Lasso witout deletion)로 구한 알고리즘이라고 이해할 수 있다.

일반화 가법 모형

- 일반화 가법 모형 (Generalized additive model, GAM)은 구조화된 비모수함수모형이다.
- 선형모형
  - $Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon.$
- 일반화 가법 모형
  - $Y = \beta_0 + \sum_{j=1}^{p} f_j(x_j) + \epsilon.$
- 예제
  - $Y = \exp(-1 + 2x_1) + \sin(2\pi x_2) + \epsilon.$

부스팅과 일반화 가법 모형

- 기저예측모형 $\mathcal{H}$가 노드가 두개인 의사결정나무로 구성되어 있다고 하자.
- Gradient 부스팅을 통하여 구축한 예측모형은 다음과 같이 쓸 수 있다:

$$F(\mathbf{x}) = \beta_0 + \sum_{T_m \in \mathcal{H}} \beta_m T_m(\mathbf{x}).$$

- $v(m)$을 나무 T_m에서 사용된 변수라고 하자.
- 그러면 앙상블 모형 F는 다음과 같이 쓸 수 있다:
  $F(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} f_j(x_j)$ 이고

$$f_j(x_j) = \sum_{T_m \in \mathcal{H}} \beta_m T_m(\mathbf{x}) I(v(m) = j)$$

  이다.
- 즉, 부스팅은 일반화 가법 모형에서 각 component 함수 f_j를 의사결정나무의 선형결합으로 추정한 것이다.

교호작용 추정

- $\mathcal{H}$가 3개의 노드를 갖는 의사결정나무로 구성되어 있다고 하자.
- 즉, 각 의사결정나무는 두개의 변수를 사용한다.
- 따라서, 앙상블 모형을 다음과 같이 쓸 수 있다 : $F(\mathbf{x}) = \beta_0 + \sum_{j,k} f_{jk}(x_j, x_k)$, 이고

$$f_{jk}(x_j, x_k) = \sum_{T_m \in \mathcal{H}} \beta_m T_m(\mathbf{x}) I\big(v(m) = \{j, k\}\big).$$

- 즉, 기저예측모형의 노드수를 결정하는 것은 앙상블모형의 교호작용의 차수를 결정하는 것이다.

해석

- 의사결정나무의 선형결합을 어떻게 해석할 수 있을까?
- 입력변수의 상대적 중요도 (Relative importance)와 부분의존도그림 (Partial dependency plot)를 이용

상대적 중요도

- 설명을 쉽게 하기 위하여, $\mathcal{H}$가 노드수가 2개인 의사결정나무로 구성되어 있다고 하자.
- 주어진 나무 T에 대해서 $s(T)$를 어미노드와 자식노드들 사이의 불순도 측정치의 차이라 하자 (불순도의 차이가 클수록 중요한 변수임).
- 변수 j의 상대적 중요도는 다음과 같이 구한다:

$$RI_j = \sum_{m=1}^{M} s(T_m) I(v(m) = j).$$

부분의존도 그림

- 주어진 앙상블 모형 $F(\mathbf{x})$에서 j번째 변수의 부분의존도 그림은 다음과 같이 정의된다:

$$f_j(x_j) = \frac{1}{n}\sum_{i=1}^{n} F\big(x_{i1}, \ldots, x_{i(j-1)}, x_j, x_{i(j+1)}, \ldots, x_{ip}\big).$$