Journal of the Korean Institute of Industrial Engineers
Vol. 42, No. 4, pp. 263-269, August 2016.
ISSN 1225-0988 | EISSN 2234-6457
http://dx.doi.org/10.7232/JKIIE.2016.42.4.263

<Original Research Paper>

대량 데이터를 위한 제한거절 기반의 회귀부스팅 기법

권혁호[1] · 김승욱[1] · 최동훈[2] · 이기천[3†]

[1]한양대학교 융합기계공학과 / [2]한양대학교 기계공학과/
[3]한양대학교 산업공학과

Boosted Regression Method based on Rejection Limits for Large-Scale Data

Hyuk-Ho Kwon[1] · Seung-Wook Kim[1] · Dong-Hoon Choi[2] · Kichun Lee[3]

[1]Dept. of Mechanical convergence Engineering, Hanyang University
[2]School of Mechanical Engineering, Hanyang University
[3]Industrial Engineering, Hanyang University

The purpose of this study is to challenge a computational regression-type problem, that is handling large-size data, in which conventional metamodeling techniques often fail in a practical sense. To solve such problems, regression-type boosting, one of ensemble model techniques, together with bootstrapping-based re-sampling is a reasonable choice. This study suggests weight updates by the amount of the residual itself and a new error decision criterion which constructs an ensemble model of models selectively chosen by rejection limits. Through these ideas, we propose AdaBoost.RMU.R as a metamodeling technique suitable for handling large-size data. To assess the performance of the proposed method in comparison to some existing methods, we used 6 mathematical problems. For each problem, we computed the average and the standard deviation of residuals between real response values and predicted response values. Results revealed that the average and the standard deviation of AdaBoost.RMU.R were improved than those of other algorithms.

Keywords: Boosting, Large Data Metamodeling, Ensemble Learning, Regression

1. 서 론

1.1 연구 필요성 및 목적

각 산업 분야에서는 오랜 기간의 연구 개발을 통해 방대한 양의 데이터와 전문 지식이 축적되어 있다. 이것을 정보로 변환하여 새로운 제품 또는 기술에 대한 연구에 적용할 수 있다면 설계 과정에서 효율성이 극대화 될 것이다.

이처럼 설계과정의 효율성을 극대화하기 위해 산업 분야에서 사용하는 방법 중 하나가 근사모델이다. 근사모델은 해석이나 실험을 통해 얻은 실제 응답 값을 기반으로 설계 공간 내에서 아직 해석이나 실험이 수행되지 않은 위치에서의 응답 값을 예측해주는 역할을 한다. 각 산업 현장에서 갖고 있는 방대한 양의 데이터를 이용하여 근사모델을 제작한다면 정확성이 높은 근사모델을 얻을 수 있다(Chen *et al*., 2001; Kodiyalam *et al*., 2004; Simpson *et al*., 2008; Wang *et al*., 2007).

하지만 일반 근사모델의 경우 방대한 데이터들을 효과적으로 다루기에는 한계가 있다. 데이터가 많아질수록 근사모델 제작에 필요한 변수 행렬이 커져 역행렬 계산이 어려워지기 때문이다. 따라서 방대한 데이터를 이용하여 한 번에 근사모델을 만드는 경우, 정확성이 부정확해진다.

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다(No. 20164010200860).
본 연구는 2015년 대한민국 교육부와 한국연구재단의 지원을 (NRF-2015S1A5A2A03047963) 받아 수행되었습니다.
† 연락저자 : 이기천 교수, 04763 서울시 성동구 왕십리로 222 한양대학교 산업공학과, Tel : 02-2220-0478, Fax : 02-2220-2299,
E-mail : skylee@hanyang.ac.kr
2016년 2월 17일 접수; 2016년 6월 2일 수정본 접수; 2016년 6월 9일 게재 확정.

이러한 문제점을 해결하고자 산업공학 분야에서 사용하는 앙상블 기법의 하나인 부스팅을 이용하고자 한다. 부스팅은 다양한 근사모델을 생성하고, 이들을 정확도에 따라 서로 다른 가중치로 조합하는 방법이다. 부스팅을 대량 데이터에 적용하는 경우에 랜덤복원추출 방법을 이용하여 방대한 데이터에서 일부 데이터를 선택한 후, 이를 이용하여 근사모델을 생성한다. 즉, 부트스트래핑 기반의 랜덤복원추출 방법으로 다양한 근사모델을 생성한 후, 정확도에 따라 단계적으로 가중치를 달리하여 조합하는 부스팅을 통해 방대한 데이터를 효과적으로 다룰 수 있다. 따라서 본 연구에서는 부스팅 기법의 대표적인 알고리즘인 AdaBoost의 개념을 이용하여 회귀모델 제작에 적합한 새로운 알고리즘을 제안하고자한다.

1.2 관련 연구

회귀모델을 위한 부스팅의 연구는 아직 미흡한 상황이다. 기존에 있던 회귀모델을 위한 부스팅 기법의 경우, 기본적으로 이미 성능이 입증된 분류를 위한 부스팅 기법을 이용하였다. 다만 이 과정에서 회귀모델 샘플을 이분 분류 샘플의 개념으로 변환하는 과정이 필요하다. 이 과정은 샘플 점에서 예측된 응답 값의 오차가 기준치보다 크면 잘못 분류된 것으로 인식되며, 크지 않으면 잘 분류된 것으로 인식되는 방법을 의미한다.

최초의 회귀모델을 위한 부스팅 기법인 AdaBoost.R의 경우, 회귀모델 샘플을 이분 분류 샘플의 개념으로 변환한 뒤 분류를 위한 AdaBoost에 적용하는 방식이다. 하지만, AdaBoost.R의 경우 실제 문제에 적용하기 어렵다는 단점을 가졌다(Freund and Schapire, 1991). AdaBoost.R2의 경우 회귀모델 나무(regression tree) 개념을 도입한 기법이었으며 AdaBoost.R의 적용 범위를 넓히기 위해 고안되었다. 하지만 수렴성이 보장되지 않는 문제가 존재하였다(Drucker, 1997). AdaBoost.RT는 상대 오차 개념을 도입한 기법으로 노이즈가 많은 데이터를 사용할 때 작동이 비교적 원활한 것으로 알려져 있으나 수렴성에 대한 이론적 근거가 부족하다(Shrestha and Solomatine, 2004; Shrestha and Solomatine, 2006). AdaBoost.SVC.R과 AdaBoost.SVR.R은 기존의 AdaBoost에 서포트 벡터 머신 개념을 이용하여 접근하였다(Kou *et al*., 2013). 이 기법의 경우 알고리즘 내에서 사용되는 파라미터 값에 대한 연구가 부족한 상황이며, 이를 최적화하기 위한 연구가 필요하다.

2. 회귀모델을 위한 새로운 부스팅 기법 제안

본 연구에서 제안한 AdaBoost.RMU.R 알고리즘의 절차는 <Figure 1>에 나와 있다. 우선 사용자가 총 6가지 값들을 결정해줘야 한다. 전체 실험점 S, 최종 앙상블 모델 생성을 위한 반복 횟수 k, 개별 근사모델 하나를 생성하기 위해 전체 실험점에서 추출되는 실험점의 개수 p, 개별 근사모델의 정확도 판단 기준 T,

Algorithm : AdaBoost.RMU.R
- create an ensemble of regression models

Input :
$S=\{(x_s, y_s)\,|\,x_s\in R^D, y_s\in R, s=1, 2, \cdots, N\}$, regression sample
k, the number of rounds
p, the number of S_i
T, threshold
f, a regression learning scheme
Z, the maximum number of model rejection per iteration

Output : A composite model

Method :
① initialize the weight w of each sample in S to $1/N$, $r_i=0$
② **for** $i=1$ to k
③ sample S_i by using sampling based on weight w with replacement and generate a model, M_i
④ compute $e_s=\dfrac{|y_s-f_i(x_s)|}{\max(y)-\min(y)}$, $m_i=\dfrac{\sum e_s}{N}$
⑤ **if** $m_i>T$ **and** $r_i<Z$ **then**
count the number of model rejection($r_i=r_i+1$) and save M_i and m_i as M_{i,r_i} and m_{i,r_i} temporarily
go back to step 3 and try again
⑥ **elseif** $m_i\le T$ or $r_i=Z$ then
⑦ **if** $r_i=Z$ **then** find the model which has the smallest m_{i,r_i} among rejected models in iteration i,
and use the m_{i,r_i} as m_i and use the corresponding model M_{i,r_i} as M_i
⑧ $mw_a=\ln\dfrac{2\cdot\max(m)-m_a}{m_a}$ and normalize $mw_a\left(\sum_{a=1}^{i}mw_a=1\right)$
⑨ compute $F(x_s)=\sum_{a=1}^{i}mw_a\cdot f_a(x_s)$, $E_s=\dfrac{|y_s-F(x_s)|}{\max(y)-\min(y)}$ and normalize $E_s\left(\sum_{s=1}^{N}E_s=1\right)$
⑩ update $w_s=w_s+E_s$ and normalize $w_s\left(\sum_{s=1}^{N}w_s=1\right)$
⑪ **endif**
⑫ **endfor**
To use the ensemble to predict response value of point, X
return $F(X)=\sum mw_i\cdot f_i(X)$ $(i=1, 2, \cdots, k)$

Figure 1. Procedure of AdaBoost.RMU.R

생성할 근사모델의 종류 f, 그리고 매 반복과정마다 모델을 반려하는 최대 회수 Z를 선택해야 한다.

전체 실험점의 가중치 w는 초기에 모두 동일하게 설정된다. i번째 반복 과정에서의 모델 반려 횟수 r_i는 모두 0으로 설정한다. 그 다음 최종 앙상블 모델을 생성하기 위한 반복과정이 실행되며 이는 k개의 개별 근사모델이 수용될 때까지 진행된다. 우선 가중치 w에 기반하여 랜덤복원추출을 수행한다. 이 때, 추출되는 데이터의 개수는 p이며 추출된 데이터를 S_i로 명명한다. S_i를 이용하여 근사모델 M_i을 생성한 후, 정확도 척도로 평균오차 m_i를 계산한다. 이를 이용하여 근사모델 M_i의 수용여부를 판단한다. 여기서 만약 m_i가 T보다 큰 경우, 근사모델 M_i는 부정확한 근사모델임을 뜻하기 때문에 M_i를 반려하게 된다.

이 때, 반려하는 횟수 r_i는 최대 Z번으로 설정하였다. 따라서 m_i가 T보다 크며 r_i가 Z보다 작을 경우, M_i를 반려하며, 이 때 M_i와 m_i를 임시적으로 저장하고 ③번 과정으로 돌아가 다시 근사모델을 생성하게 된다. 하지만 m_i가 T보다 작거나, r_i가 Z인 경우에는 다음의 과정이 진행된다. 우선 m_i가 T보다 작은 경우, 근사모델 M_i를 정확한 모델로 판단하여 현재 반복과정에서의 개별 근사모델로 수용하게 된다. 그리고 아직 m_i가 T보다 크지만 r_i가 Z인 경우에는 임시적으로 저장된 반려 모델 Z개 중 m_i가 가장 낮은 모델을 선택하여 그 모델을 수용하게 된다. 이 후에는 현재 반복 과정까지 수용되었던 모든 개별 근사모델들에 대한 정확도에 기반하여 각각의 모델 가중치 mw_a를 계산하며. 이어서 mw_a의 합이 1이 되도록 스케일링을 진행한다. 이 때, mw_a와 m_a는 각각 a번째 모델 가중치와 정확도 척도를 의미한다. 수용된 모든 개별 근사모델과 각각의 모델 가중치를 이용하여 현재 만들 수 있는 앙상블 모델을 제작하고, 이를 이용하여 전체 실험점(S)에서의 응답값을 예측한다. 그리고 예측된 응답값과 실제 응답값 사이의 오차를 계산하고, 계산된 오차의 전체 합이 1이 되도록 스케일링을 진행한다. 이 때, $f_a(x_s)$는 a 번째의 개별 근사모델에 의해 예측되는 x_s에서의 예측값이다. 이를 이용하여 w_s의 업데이트를 진행한다. 업데이트 후, w_s의 합이 1이 되도록 스케일링 한다. 이 때, w_s는 s번째의 실험점이 갖는 가중치를 의미한다.

위와 같은 과정을 최종 앙상블 모델을 생성하기까지 k번 반복하며, 반복이 종료되면 최종 앙상블 모델이 얻어진다.

3. 제안한 알고리즘의 성능 평가방법

AdaBoost.RMU.R의 성능 확인을 위해 비교 모델로 회귀모델을 위한 bootstrap aggregating(bagging)과 AdaBoost.RT 2가지를 사용한다. 예제로는 Branin, SK7, RSB(Madsen and Zilinskas, 2000), Mystery(Sasena, 2002), Linear, Quad 6개의 수학 예제를 사용하며, <Figure 2>에 소개되어 있다.

• Problem : Branin

$$y=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos(x_1)+10$$

Lower bound = [-5 0], Upper bound = [10 15]

• Problem : Shekel7(SK7)

$$y=-\sum_{i=1}^{7}\left(\sum_{j=1}^{4}(x_j-C_{ji})^2+\beta_i\right)^{-1},$$

$where$

$$C=\begin{bmatrix}4&1&8&6&3&2&5\\4&1&8&6&7&9&5\\4&1&8&6&3&2&3\\4&1&8&6&7&9&3\end{bmatrix},\ \beta=[0.1\ 0.2\ 0.2\ 0.4\ 0.4\ 0.6\ 0.3]$$

Lower bound = [0 0 0 0], Upper bound = [10 10 10 10]

• Problem : Mystery

$$y=2+0.01(x_2-x_1^2)^2+(1-x_1)^2+2(2-x_2)^2+7\sin(0.5x_1)\sin(0.7x_1x_2)$$

Lower bound = [0 0], Upper bound = [5 5]

• Problem : Linear

$$y=2.51+0.523x_1+1.396x_2+8.054x_3+4.83x_4+20.8346x_5$$

Lower bound = [0 0 0 0 0], Upper bound = [10 10 10 10 10]

• Problem : Quad

$$y=2.51+0.523x_1+1.396x_2+8.054x_3+4.83x_4+20.8346x_5+12.085x_1^2+5.18x_2^2+4.843x_3^2+2.084x_4^2+9.481x_5^2+9.3232x_1x_2+15.8138x_1x_3+1.1883x_1x_4+9.8423x_1x_5+7.8553x_2x_3+28.8545x_2x_3+11.813x_2x_5+8.0546x_3x_4+14.8513x_3x_5+12.812x_4x_5$$

Lower bound = [0 0 0 0 0], Upper bound = [10 10 10 10 10]

• Problem : Rosenbrock(RSB)

$$y=\sum_{i=1}^{n-1}100(x_{i+1}-x_i^2)^2+(1-x_i)^2$$

$where$

$X=[x_1,\cdots,x_n]\in R^n$, $n=5$.

Lower bound = [0 0 0 0 0], Upper bound = [10 10 10 10 10]

Figure 2. Problems Used to Estimate Performance of the Algorithm

Table 1. Input Data for each Problem of AdaBoost.RT

	the number of rounds(k)	the number of $S_i(p)$	threshold (ϕ)	the number of regression samples(S)	regression learning scheme
Branin	200	500	0.013	20000	Radial basis function interpolation (Powell, 1987)
SK7			0.22		
Mystery			6.84×10^{-4}		
Linear			1.33×10^{-16}		
Quad			0.10		
RSB			0.15		

Table 2. Input Data for each Problem of AdaBoost.RMU.R

	the number of rounds(k)	the number of $S_i(p)$	threshold (T)	the number of regression samples(S)	regression learning scheme
Branin	200	500	5.12×10^{-4}	20000	Radial basis function interpolation (Powell, 1987)
SK7			0.017		
Mystery			1.93×10^{-4}		
Linear			7.22×10^{-17}		
Quad			0.019		
RSB			0.023		

AdaBoost.RMU.R과 AdaBoost.RT에서 사용자가 결정해줘야 하는 공통 입력값 5가지를 <Table 1>, <Table 2>와 같이 설정한다. 여기서 threshold 값의 결정 방법은 각 수학 예제에 대해 임의 추출을 통한 모델 생성을 50번 수행한 후, 여기서 얻어지는 상대오차를 AdaBoost.RT의 threshold 값으로 결정하였으며 AdaBoost.RMU.R의 경우는 절대오차를 y의 최대값과 최소값의 차이로 스케일링하여 threshold 값으로 선정하였다. 이처럼 threshold 값을 다르게 선정한 이유는 AdaBoost.RMU.R과 AdaBoost.RT의 오차 판별식이 다르기 때문이다.

성능 측정 방법은 다음과 같다. 우선 근사모델을 만드는데 사용한 실험점들과 별개로 성능 평가를 위한 시험점을 따로 1,000개 추출한 후, 해당 시험점에서의 실제 응답과 최종 앙상블 모델로 얻어진 예측값 사이의 오차를 구한다. 이 때, 오차는 절대 오차를 y의 최대값과 최소값의 차이로 스케일링한 값이다. 그리고 1,000개의 오차값들에 대한 평균과 표준편차를 이용하여 정확성을 평가하며 최종 앙상블 모델을 만들기까지 필요한 개별 근사모델의 개수로 효율성을 평가한다. 또한 랜덤성을 고려하여 각 문제에 대해 20번의 반복 수행을 실시하여 3가지 알고리즘의 성능을 비교한다.

4. 결과 및 분석

AdaBoost.RMU.R에서는 <Table 1>, <Table 2>에 나와 있는 입력값 외에 추가적으로 최대 모델 반려 횟수 Z를 결정해줘야 한다. 본 연구에서는 이 값을 결정하기 위해, Z를 3, 4, 5로 변화시켜가며 AdaBoost.RMU.R의 정확도와 효율성을 평가하였다. <Table 3>, <Table 4>, <Table 5>는 각각 최대 모델 반려 횟수 Z를 3, 4, 5로 변화시켰을 때, 예제에 따른 오차의 평균, 표준편차, 개별 근사모델 생성 횟수를 나타낸다. 세 Table의 값은 각각 반려 횟수가 4인 경우를 기준으로 스케일링하였다. <Table 3>, <Table 4>를 통해 최대 모델 반려 횟수를 3, 4, 5로 변화시킴에도 불구하고 예제에 따른 오차의 평균과 표준편차 값은 큰 차이가 없다는 것을 알 수 있다.

하지만 개별 근사모델 생성 횟수의 경우, 최대 모델반려 횟수가 4일 때, 가장 적은 것을 <Table 5>를 통해 알 수 있다. 따라서 본 연구에서는 최대 모델 반려 횟수 Z를 4로 설정하였다.

Table 3. Relative Value of Mean of Error at Test Points

	3	4	5
Branin	1.00	1.00	0.96
SK7	1.00	1.00	1.00
Mystery	1.07	1.00	0.99
Linear	1.03	1.00	1.17
Quad	1.00	1.00	1.00
RSB	1.00	1.00	1.00

Table 4. Relative Value of Standard Deviation of Error at Test Points

	3	4	5
Branin	0.94	1.00	0.99
SK7	1.14	1.00	1.05
Mystery	1.07	1.00	0.98
Linear	1.04	1.00	1.10
Quad	1.01	1.00	1.01
RSB	1.00	1.00	1.00

Table 5. Relative Value of the Number of Generation of Individual Models

	3	4	5
Branin	0.98	1.00	1.01
SK7	1.51	1.00	1.32
Mystery	0.98	1.00	1.01
Linear	1.01	1.00	1.03
Quad	1.09	1.00	1.44
RSB	1.84	1.00	2.28

<Table 6>과 <Table 7>은 bagging을 기준으로 하여 1,000개의 시험점에 대한 오차의 평균과 표준편차를 각 알고리즘에 대해 상대적으로 나타낸 표이다. 따라서 bagging의 값 1을 기준으로 숫자가 작을수록 더 정확한 것이다.

<Table 6>과 <Table 7>을 통해 AdaBoost.RMU.R이 bagging에 비해 RSB 문제만 동일한 정확도를 보이고 나머지 문제에 대해서는 더 우수한 것을 알 수 있다. 정량적으로 AdaBoost.RMU.R이 bagging에 비해 오차의 평균은 평균 41.5% 개선되었으며, 오차의 표준편차도 평균 53.8% 향상되었다.

AdaBoost.RT에 비해서는 오차의 평균이 SK7, Linear 문제에서만 동일한 정확도를 보이고 나머지 문제들에 대해서는 AdaBoost.RMU.R이 더 좋은 정확도를 보였다. 또한 오차의 표준편차는 모든 문제에 대해서 AdaBoost.RMU.R이 더 우수한 결과를 보였다. AdaBoost.RMU.R은 AdaBoost.RT에 비해 오차의 평균이 평균 35.0% 개선되었으며 오차의 표준편차는 평균 39.5% 향상되었다. 따라서 정확도 측면에서는 AdaBoost.RMU.R이 기존 알고리즘인 bagging, AdaBoost.RT에 비해 우수한 것을 알 수 있다.

Table 6. Relative Value of Mean of Error at Test Points

	bagging	AdaBoost.RT	AdaBoost.RMU.R
Branin	1.00	23.53	0.12
SK7	1.00	0.95	0.95
Mystery	1.00	15.42	0.27
Linear	1.00	0.18	0.18
Quad	1.00	1.14	0.99
RSB	1.00	1.06	1.00

Table 7. Relative Value of Standard Deviation of Error at Test Points

	bagging	AdaBoost.RT	AdaBoost.RMU.R
Branin	1.00	4.68	0.03
SK7	1.00	0.68	0.55
Mystery	1.00	3.62	0.07
Linear	1.00	0.22	0.21
Quad	1.00	0.99	0.91
RSB	1.00	1.15	1.00

이러한 결과의 요인 중 하나는 샘플링 방법의 차이라 판단된다. 우선 bagging의 경우, 각 실험점들이 동일한 가중치를 갖으며 이 값을 기반으로 실험점들이 랜덤하게 선택되었다. 앙상블 모델의 정확도를 향상시키기 위해서는 오차가 큰 실험점에서 샘플링이 수행되어야 하지만 bagging은 오차의 크기와는 관계없이 모든 실험점이 동일한 가중치를 갖기 때문에 다른 기법들에 비해 오차가 큰 실험점이 선정될 확률이 상대적으로 낮아지게 되는 것이다. 그리고 AdaBoost.RT의 경우, bagging과 달리 전체 실험점에 배당되는 가중치를 매 반복과정마다 갱신하게 된다. 하지만 이전 반복 과정에서 생성된 개별 근사모델을 기준으로 가중치를 갱신하게 된다. 따라서 이 방법은 이전 개별 근사모델을 기반으로 한 제한된 향상 방향의 가중치 갱신이라 볼 수 있다. 또한 AdaBoost.RT는 예측 오차가 특정한 기준을 만족시키지 못하면 기존 가중치를 유지시키고 만족하는 경우에는 모두 동일하게 가중치를 갱신시켜 버린다. 따라서 오차의 크기를 전혀 고려하지 못하고 가중치를 갱신하는 단점이 있다. 하지만 본 연구에서 제안한 AdaBoost.RMU.R의 경우, 현재까지 승인되었던 모든 개별 근사모델을 이용하여 현재 만들 수 있는 앙상블 모델을 생성하고, 이를 이용한 오차를 기반으로 각 실험점들의 가중치를 갱신시켰다. 이로 인해 앙상블 모델의 정확도 향상을 위해 필요한 위치에서 샘플링이 수행되었다. 또한 오차의 크기에 따라 서로 다른 값으로 가중치를 갱신하기 때문에 오차가 큰 실험점이 선택될 가능성이 높아지며, 이는 결국 최종 앙상블 모델의 정확도 향상으로 이어졌다.

AdaBoost.RMU.R이 정확도 측면에서 bagging, AdaBoost.RT에 비해 우수한 결과를 보인 두 번째 요인은 오차 측정 방법에 있다고 판단된다. AdaBoost.RT의 경우, 상대오차 개념을 이용하여 각 실험점에서의 오차를 계산하게 된다. 이러한 경우, 실제 응답값이 0 근방일 때 상대오차를 계산하는 과정에서 실제 오차가 크지 않음에도 실제 응답값이 0에 가깝다보니 상대오차가 굉장히 커지는 경우가 발생하게 된다.

Table 8. Relative Value of the Number of Generation of Individual Models

	bagging	AdaBoost.RT	AdaBoost.RMU.R
Branin	1.00	1.00	1.08
SK7	1.00	1.00	1.08
Mystery	1.00	1.00	1.09
Linear	1.00	1.00	1.55
Quad	1.00	1.00	2.11
RSB	1.00	1.00	1.77

결국 정확한 오차를 측정할 수 없어 각 실험점에 대한 가중치를 올바르게 갱신할 수 없다. 이러한 문제점을 해결하기 위해 AdaBoost.RMU.R의 경우 절대 오차를 y의 최대값과 최소값의 차이로 스케일링하였다. 따라서 각 실험점들이 보다 논리적인 가중치를 부여받게 되었다. 이는 결국 최종 앙상블 모델의 정확도 향상으로 이어지게 되었다. bagging의 경우는 항상 모든 실험점이 동일한 가중치를 갖기 때문에 특별히 각 실험점에서의 오차를 측정하지 않는다.

<Table 8>은 bagging을 기준으로 하여 최종 앙상블 모델을 만들기까지 필요한 개별 근사모델의 개수를 각 알고리즘에 대해 상대적으로 나타낸 표이다. <Table 8>을 통해 본 연구에서 제안한 AdaBoost.RMU.R이 bagging, AdaBoost.RT에 비해 모든 문제에 대하여 더 많은 개별 근사모델을 필요로 하는 것을 알 수 있다. bagging과 AdaBoost.RT의 경우 1회의 반복 과정동

안 하나의 개별 근사모델을 생성하게 된다. 하지만 AdaBoost.RMU.R의 경우 정확도가 높은 개별 근사모델로 최종 앙상블 모델을 조합하기 위해 1회 반복 과정에서 최대 4개의 근사모델을 생성하게 된다. 따라서 bagging과 AdaBoost.RT에 비해 AdaBoost.RMU.R이 최종 앙상블 모델을 만들기까지 평균 1.45배 더 많은 개별 근사모델을 필요로 하게 되었다.

5. 결 론

대량 데이터를 모두 이용하여 일반 근사모델을 만드는 경우 수치적인 한계로 인해 근사모델이 부정확해진다. 따라서 본 연구에서는 이러한 일반 근사모델이 갖는 대량 데이터 처리 한계점을 극복하고자 하였다. 랜덤 복원 추출 방법을 통해 대량 데이터에서 일부 데이터를 이용하여 제한거절 방법 기반으로 근사모델을 생성하고, 이와 같은 과정을 반복하여 다양한 근사모델을 생성한 후 이들을 정확도에 따라 가중치를 달리하여 조합함으로써 효과적으로 대량 데이터를 다룰 수 있는 AdaBoost. RMU.R을 제안하였다.

AdaBoost.RMU.R의 경우 bagging과 AdaBoost.RT에 비해 최종 근사모델을 만들기까지 평균 1.45배 많은 개별 근사모델을 필요로 했다. 하지만 bagging에 비해 오차의 평균이 평균 41.5% 향상되었으며, 오차의 표준편차도 평균 53.8% 개선되었다. 또한 AdaBoost.RT에 비해 오차의 평균이 평균 35.0% 개선되었으며 오차의 표준편차는 평균 39.5% 향상되었다. 이러한 결과를 얻을 수 있었던 AdaBoost.RMU.R의 특징 즉, 이전 알고리즘과의 차이는 다음과 같이 요약할 수 있다.

1. 기존 알고리즘은 1회의 반복 과정 중 하나의 개별 근사모델을 생성하였으나 AdaBoost.RMU.R의 경우 더 정확성 높은 개별 근사모델을 사용하여 최종 앙상블 모델을 생성하기 위해 1회의 반복 과정에서 최대 4개의 개별 근사모델을 생성하도록 하였다.
2. Bagging의 경우 각 실험점에서의 오차는 고려하지 않고 항상 동일한 가중치로 샘플링이 수행되기 때문에 다른 기법들에 비해 오차가 큰 실험점이 선택될 확률이 상대적으로 낮아지게 된다. AdaBoost.RT는 이전에 생성되었던 개별 근사모델의 오차를 이용하여 가중치를 갱신하고, 이를 기반으로 샘플링을 수행했다. 따라서 이 방법은 이전 개별 근사모델을 기반으로 한 제한된 향상 방향의 가중치 갱신이라 볼 수 있다. 또한 가중치를 갱신하는 과정에서 오차의 크기를 고려하지 못하고 미리 정해놓은 예측 오차 기준을 이용하여 만족시키는 경우, 동일하게 가중치를 갱신시키고 만족하지 못하는 경우에는 기존과 동일하게 가중치를 유지시켰다. 하지만 AdaBoost.RMU.R의 경우 현재 생성할 수 있는 앙상블 모델의 오차를 기반으로 샘플링을 수행하기 때문에 최종 예측값을 기반으로 한 가중치를 부여하게 된다. 이는 결국 최종 앙상블 모델의 정확도를 향상시키기 위해 필요한 위치에서 샘플링이 수행되도록 하였다. 또한 모든 실험점에서 얻어지는 오차를 기반으로 각각 다르게 가중치를 갱신시키기 때문에 보다 정확한 확률로 실험점이 선정되었다. 결국 이를 통해 최종 앙상블 모델의 정확도를 향상시킬 수 있었다.
3. Bagging의 경우 모든 실험점이 항상 동일한 가중치로 선정되기 때문에 특별히 각 실험점에서의 오차를 측정하지 않았다. AdaBoost.RT의 경우는 상대오차 개념을 이용하여 각 실험점에서의 오차를 측정하기 때문에 절대 오차가 크지 않음에도 불구하고 y의 실제값이 0 근방인 경우 오차가 크게 측정되었다. 이로 인해 이러한 실험점들에서 가중치가 커지게 되고, 결국 샘플링도 자주 수행되었다. 하지만 AdaBoost.RMU.R의 경우, 절대 오차를 y의 최대값과 최소값의 차이로 스케일링한 값을 이용하여 오차를 측정하기 때문에 AdaBoost.RT에서 발생하는 문제점을 해결하였다. 이는 통해, 올바른 가중치로 샘플링을 수행하게 되었다.

AdaBoost.RMU.R이 기존 알고리즘에 비해 상당 부분 개선이 있었으나 아직 부족한 점도 있다. 수학적 예제로 사용했던 RSB의 경우 AdaBoost.RMU.R이 더 많은 개별 근사모델을 필요로 했으나 오차의 평균과 표준편차는 bagging, AdaBoost.RT와 동일하였다. 또한 Quad 문제의 경우 AdaBoost.RMU.R이 개별 근사모델을 bagging, AdaBoost.RT에 비해 2배 이상 필요로 하였으나 오차의 평균, 표준편차의 개선 정도는 미미하였다. 따라서 향후, AdaBoost.RMU.R을 더욱 개선하여 RSB, Quad 문제에서도 확실한 개선을 보여야 할 것으로 판단된다.

참고문헌

Drucker, H. (1997), Improving regressors using boosting techniques, Proceedings of the 14th International Conference of Machine Learning.

Freund, Y. and Schapire, R. E. (1997), A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of computer and system sciences*, **55**(1), 119-139.

Gao, F., Kou, P., Gao, L., and Guan, X. (2013), Boosting regression methods based on a geometric conversion approach : using SVMs base learners, *Neurocomputing*, **113**(3), 67-87.

Jin, R., Chen, W., and Simpson, T. W. (2001), Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria, *Structural and Multidisciplinary Optimization*, **23**(1), 1-13.

Kodiyalam, S., Yang, R. J., and Gu, L. (2004), High-Performance Computing and Surrogate Modeling for Rapid Visualization with Multidisciplinary Optimization, *AIAA Journal*, **42**(11), 2347-2354.

Madsen, K. and Zilinskas, J. (2000), Testing branch-and-bound methods for global optimization, IMM technical report, Technical University of Denmark.

Sasena, M. J. (2002), Flexibility and Efficiency Enhancement for Constrained Global Design Optimization with Kriging Approximations, PhD thesis, University of Michigan.

Park, C. I., Kim, Y. D., Kim, J. S., Song, J. W., and Choi, H. S. (2011), Data Mining with R, Kyowoosa.

Powell, M. J. D. (1987), Radial Basis Functions for Multivariable Interpolation : A review, Oxford University Press, 143-167.

Shrestha, D. L. and Solomatine, D. P. (2006), Experiments with Ada Boost.RT : an improved boosting scheme for regression, *Neural computation*, **18**(7), 1678-1710.

Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C. (2008), Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization : A Review of How Far We Have Come-or Not, 12th AIAA/ISSMO Multidisciplinary and Optimization Conference.

Solomatine, D. P. and Shrestha, D. L. (2004), AdaBoost.RT : a boosting algorithm for regression problems, Proceedings of the International Joint Conference on Neural Networks.

Wang, G. G. and Shan, S. (2007), Review of Metamodeling Techniques in Support of Engineering Design Optimization, *Journal of Mechanical Design*, **129**(4), 370-380.