

범죄율 예측을 위한 간단한 모델

목차

1

서론

- 분석 주제 및 목적
- 데이터 설명 및 전처리
- 분석 방법론 소개

2

분석

- 변수선택 : LASSO
- 차원축소 : SIR

3

결론

- 예측 성능 확인 및 비교
- 모델 해석
- 분석의 결론, 의의, 한계

1-1. 분석 주제 및 목적

분석 주제

- 여러 지표를 사용해 범죄율을 예측하는 모델 만들기

분석 목적

- 범죄율에 영향을 미치는 주요 요인을 분석 → 단순하고 해석이 가능한 모델
- 범죄율을 최대한 정확히 예측해 범죄 예방에 도움 → 예측 성능이 관측은 모델

1-2. 데이터 설명 및 전처리

분석에 사용한 데이터

- Kaggle의 **Communities and Crime data**
- $n = 2215$, $p = 147$ (id = 4, predictor = 125, target = 18)
- 각 관측치는 미국의 community를 의미 (community \subset county \subset state)

1-2. 데이터 설명 및 전처리

target variables (18개)

- 살인, 강간, 강도, 폭행, 빈집털이, 자동차 털이, 절도, 방화의 8개 범죄에 대한
범죄 건수와 10만명 당 범죄 건수
- 10만명 당 강력 범죄(살인, 강간, 강도, 폭행) 건수 ➡ 분석에 사용한 반응변수
- 10만명 당 비폭력 범죄(빈집털이, 자동차 털이, 절도, 방화) 건수

predictors (125개)

- 인구, 인종 비율, 이민자 비율 등 사회 전반에 관련된 지표들

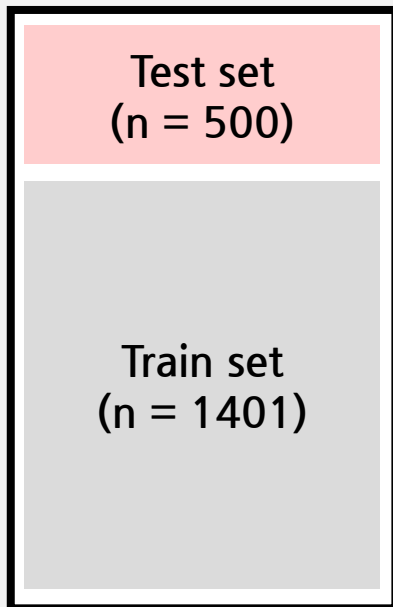
1-2. 데이터 설명 및 전처리

전처리

- 결측치 제거
- 반응변수(10만명 당 강력 범죄 건수)를 반올림해서 자연수로 변환

➔ 최종적으로 $n = 1901$, $p = 103$ (predictors = 102, target = 1)

Train/Test split



- 설명변수 정규화
- 10-fold split
(hyperparameter tuning, cross validation error)

1-3. 분석 방법론 소개

분석에 사용한 방법론

- 102개의 설명변수를 모두 사용할 경우 모델이 복잡함

➡ LASSO 변수 선택 / SIR 차원축소 방법을 사용해 단순한 모델

- 해석이 가능한 모델, 예측 성능도 고려

➡ Generalized Linear Model(GLM)과 Generalized Additive Model(GAM)

반응변수에 대한 Poisson distribution 가정

square root link function 사용

2-1. LASSO

변수선택을 통한 단순한 모델 적합

- **LASSO** : l_1 penalty를 사용한 shrinkage 방법

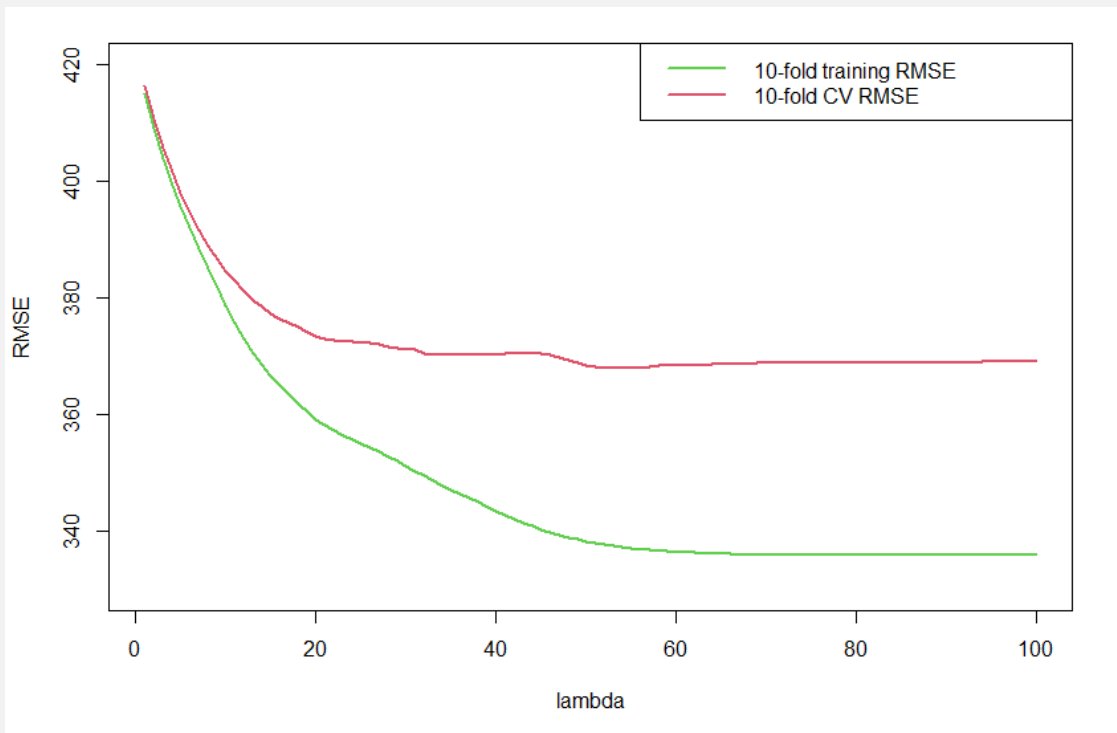
$$\hat{\beta}_{\lambda}^L = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- hyperparameter : $\lambda \geq 0$, λ 값이 클수록 shrinkage가 크게 일어남
- LASSO regression은 회귀계수를 0으로 shrink하므로 **변수 선택**의 기능

2-1. LASSO

LASSO를 적용한 GLM 적합

- parameter tuning by grid search
: 10-fold cross validation으로 tuning (λ : 10 ~ 0.0001까지 100개)
- parameter tuning 결과



- 54번째 λ (= 0.0210)일 때
가장 작은 CV RMSE(=367.8678)
- Train set 전체로 해당 모델을 적합했을 때
16개 변수의 회귀계수가 0으로 shrink되어
해당 모델에 사용된 변수는 86개
➡ 아직 설명변수가 많아서 모델이 복잡함

2-1. LASSO

추가적인 변수선택

| λ | $\lambda_1 = 10$ | $\lambda_2 = 8.90$ | $\lambda_3 = 7.92$ | $\lambda_4 = 7.05$ | $\lambda_5 = 6.28$ | $\lambda_6 = 5.59$ | $\lambda_7 = 4.98$ | $\lambda_8 = 4.43$ | $\lambda_9 = 3.94$ |
|-----------|------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| p | 3 | 4 | 4 | 4 | 4 | 6 | 8 | 8 | 9 |

- Training set으로 적합해본 결과, 첫 9개 λ 값에서 10개 이하의 설명변수를 사용한 모델이 적합된 것을 확인

| | |
|--|--|
| λ_1 | <ul style="list-style-type: none">• racePctWhite(백인 인구 비율), PctKids2Par(양부모 가정에 속한 아동 비율)• PctKidsBornNeverMar(미혼 가정에 속한 아동 비율) |
| $\lambda_2, \lambda_3, \lambda_4, \lambda_5$ | <ul style="list-style-type: none">• FemalePctDiv(여성 이혼율) |
| λ_6 | <ul style="list-style-type: none">• PctPersDenseHous(거주 인구 대비 좁은 집의 비율), TotalPctDiv(전체 이혼율) |
| λ_7, λ_8 | <ul style="list-style-type: none">• MalePctDivorce(남성 이혼율), HousVacant(빈 집의 수) |
| λ_9 | <ul style="list-style-type: none">• MedRentPctHousInc(가구 소득에서 임대료가 차지하는 비율의 중앙값) |

2-1. LASSO

- 해당 변수들을 사용해 **GAM** 적합, 10-fold cross validation으로 적절한 설명변수의 수를 선택
- GAM에서 각 설명변수의 basis function은 effective $df = 4$ 의 smoothing splines과 $df = 4$ 의 natural cubic splines 사용
- 10-fold cross validation 결과 (10-fold CV RMSE)

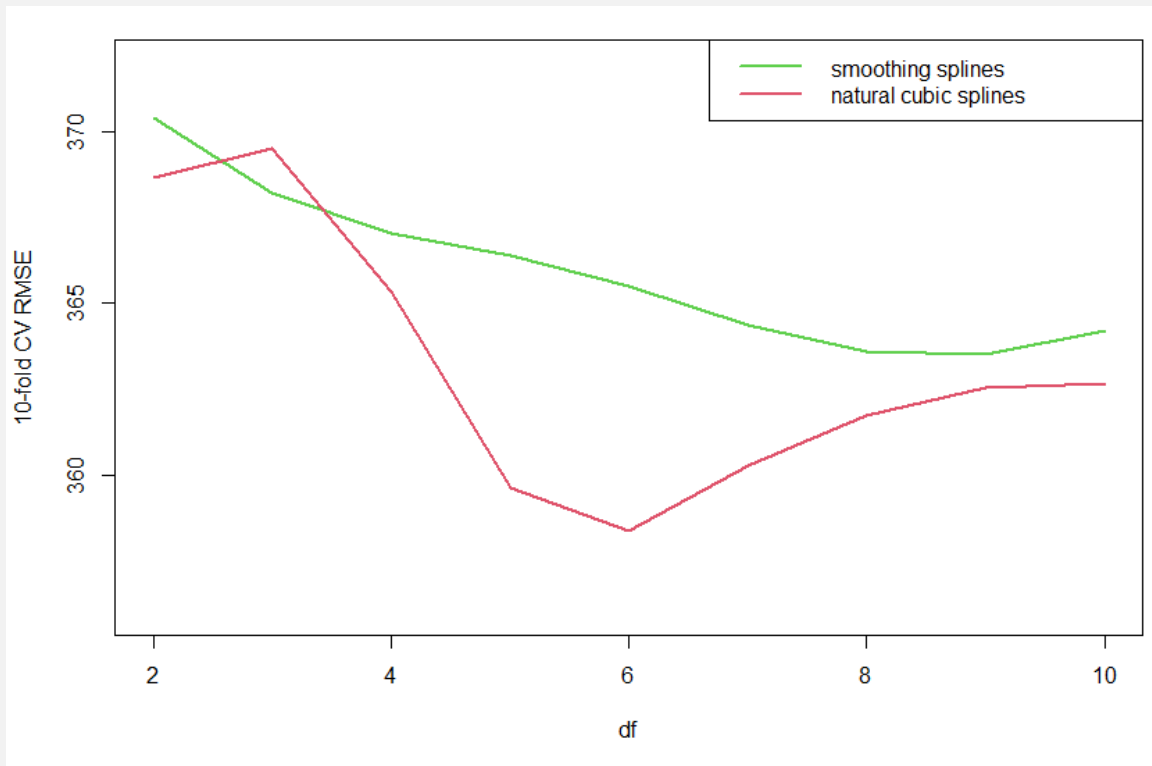
| | smoothing splines (effective $df = 4$) | natural cubic splines ($df = 4$) |
|---------|---|------------------------------------|
| $p = 3$ | 383.0747 | 384.6935 |
| $p = 4$ | 376.6479 | 379.4138 |
| $p = 6$ | 367.0452 | 371.4940 |
| $p = 8$ | 437.2445 | 365.3163 |
| $p = 9$ | 444.0325 | 367.5477 |

- LASSO GLM에서 가장 좋은 성능을 보였던 86개의 설명변수를 사용한 모델(CV RMSE = 367.8678)과 비슷하거나 더 좋은 성능을 보임 ➡ **성공적인 차원축소**

2-1. LASSO

hyperparameter tuning

- smoothing spline : effective df
 - natural cubic spline : df
- } basis function의 flexibility에 관여 ➡ tuning parameter



- effective df, df 모두 2~10의 값을 사용해 10-fold cross validation을 통한 grid search
- smoothing splines은 effective df = 9일 때 CV RMSE = 363.5151로 가장 좋은 성능
- natural cubic splines은 df = 6일 때 CV RMSE = 358.3709로 가장 좋은 성능
- 대체로 natural cubic splines의 성능이 좋게 나타나 **natural cubic splines 모델을 선택해 모델 개선 진행**

2-1. LASSO

모델 개선(1) : GAM with natural cubic splines (df = 6)

| | |
|--------------|--|
| 핵심 설명변수 | racePctWhite(백인 인구 비율), PctKids2Par(양부모 가정에 속한 아동 비율), PctKidsBornNeverMar(미혼 가정에 속한 아동 비율) |
| 이혼과 관련된 설명변수 | FemalePctDiv(여성 이혼율), MalePctDivorce(남성 이혼율), TotalPctDiv(전체 이혼율) |
| 집과 관련된 설명변수 | PctPersDenseHous(거주 인구 대비 좁은 집의 비율), HousVacant(빈 집의 수) |

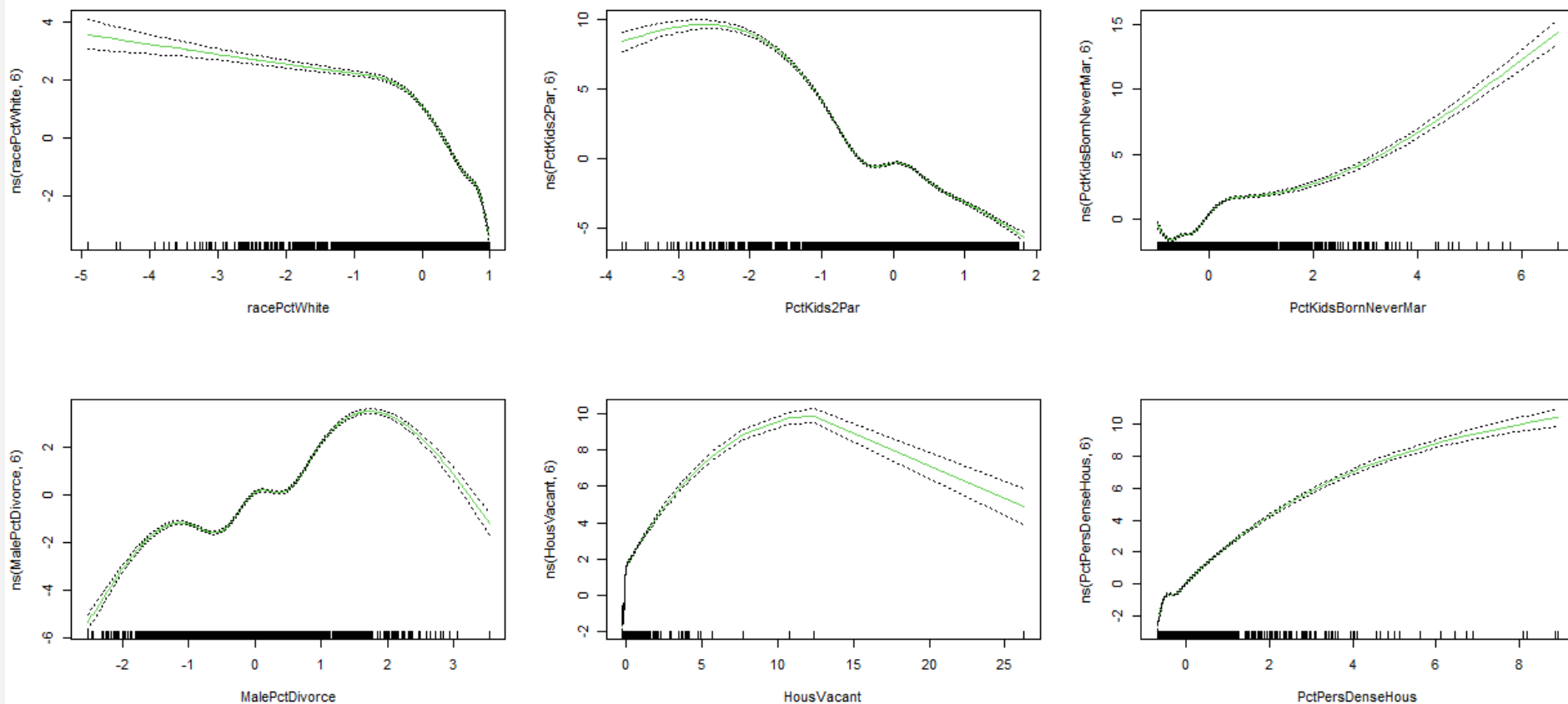
- 이혼과 관련된 설명변수와 집과 관련된 설명변수를 줄여서 더 간단한 모델을 만들고자 함
- 각 변수들의 여러 조합을 확인해본 결과,

racePctWhite(백인 인구 비율), PctKids2Par(양부모 가정에 속한 아동 비율),
PctKidsBornNeverMar(미혼 가정에 속한 아동 비율), MalePctDivorce(남성 이혼율),
PctPersDenseHous(거주 인구 대비 좁은 집의 비율), HousVacant(빈 집 수)

6개 변수를 사용한 모델의 10-fold CV RMSE가 357.7633로 가장 좋은 성능을 보임
(8개의 변수 모두를 사용한 모델의 10-fold CV RMSE는 358.3709로 성능이 향상되었음)

2-1. LASSO

모델 개선(2) : GAM with natural cubic splines (df = 6)



- training set에 모델을 적합 후 GAM plot을 참고해 변수마다 basis function 및 df 조정

2-1. LASSO

최종적으로 선택된 GAM

| 설명변수 | Basis function |
|---------------------------------------|-----------------------------------|
| racePctWhite (백인 비율) | natural cubic splines with df = 3 |
| PctKids2Par (양부모 가정에 속한 아동 비율) | natural cubic splines with df = 5 |
| PctKidsBornNeverMar (미혼 가정에 속한 아동 비율) | Identity (= linear model) |
| MalePctDivorce (남성 이혼율) | natural cubic splines with df = 6 |
| HousVacant (빈 집의 수) | natural cubic splines with df = 6 |
| PctPersDenseHous (거주 인구 대비 좁은 집의 비율) | natural cubic splines with df = 2 |

86개의 설명변수를 사용한 LASSO GLM의 10-fold CV RMSE = 367.8678

➡ 최종적으로 선택된 GAM의 10-fold CV RMSE = **356.2700**

➡ 변수선택을 통한 차원축소 및 성능 향상

2-2. SIR (1)

차원축소를 통한 단순한 모델 적합

- Supervised dimension reduction

: find β s.t. $Y = f(X_1, \dots, X_p) + \epsilon \Leftrightarrow Y = g(\beta_1^T X, \dots, \beta_d^T X) + \epsilon$ (where $d \ll p$)

- Inverse Regression

: eigen decomposition to $Var[E(X|Y)] \rightarrow \beta$ consist of each eigenvector (as like PCA)

select d eigen vector

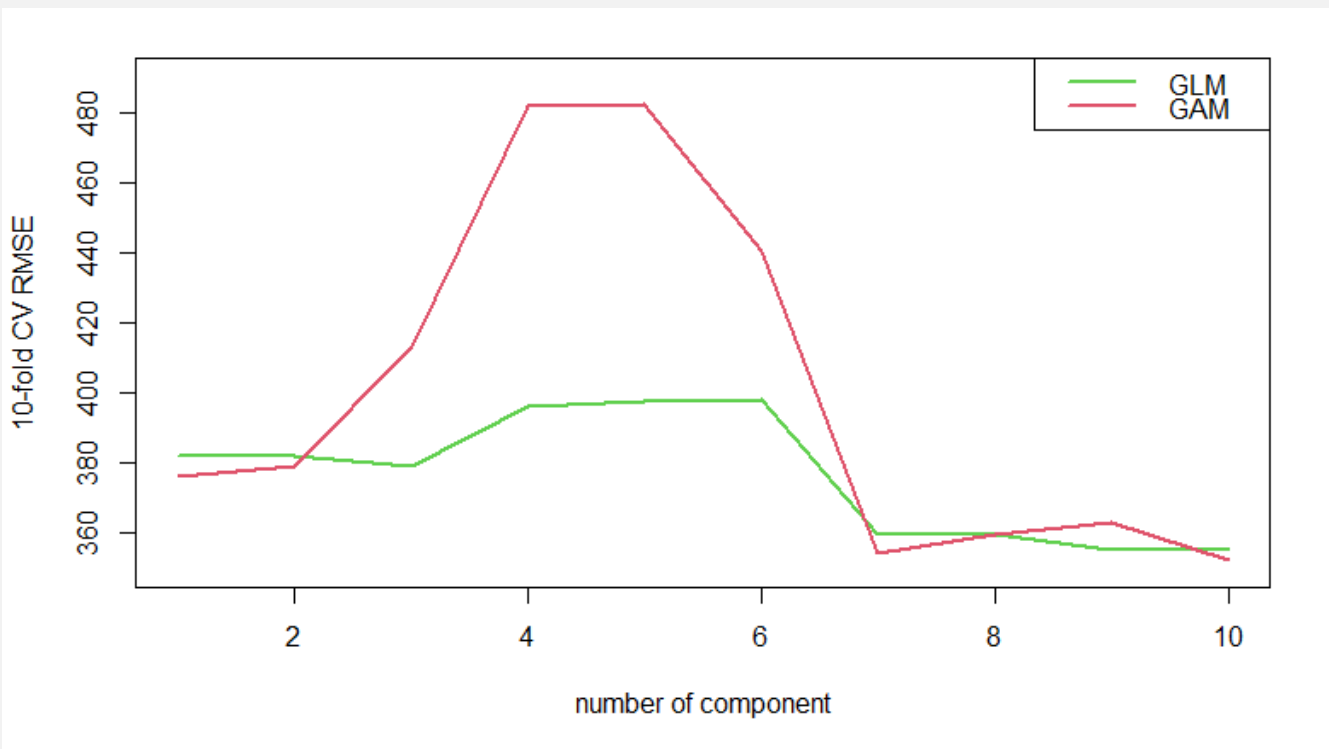
- SIR(Sliced Inverse Regression)

: slicing Y because of practical issue

2-2. SIR (1)

SIR 차원축소 진행

- 1~10개의 eigenvector를 선택해 10-fold cross validation error 확인
- 사용한 모델 : GLM, GAM(각 변수의 basis function은 effective df=4인 smoothing spline)

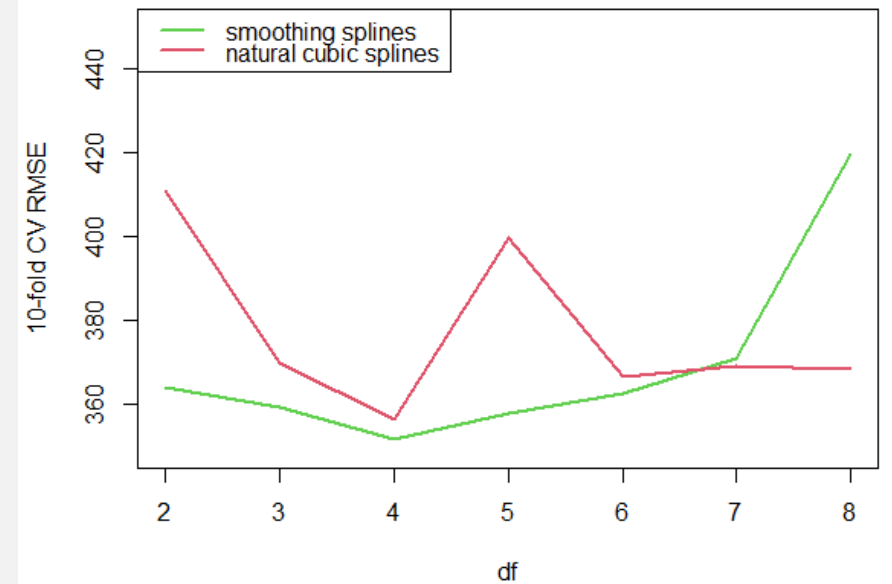


- GLM은 9개의 component를 사용했을 때
CV RMSE = 355.0469로 가장 성능이 좋음
- GAM은 10개의 component를 사용했을 때
CV RMSE = 351.8112로 가장 성능이 좋음
- GAM은 3~6개의 component를 사용했을 때
성능이 매우 좋지 않음

2-2. SIR (1)

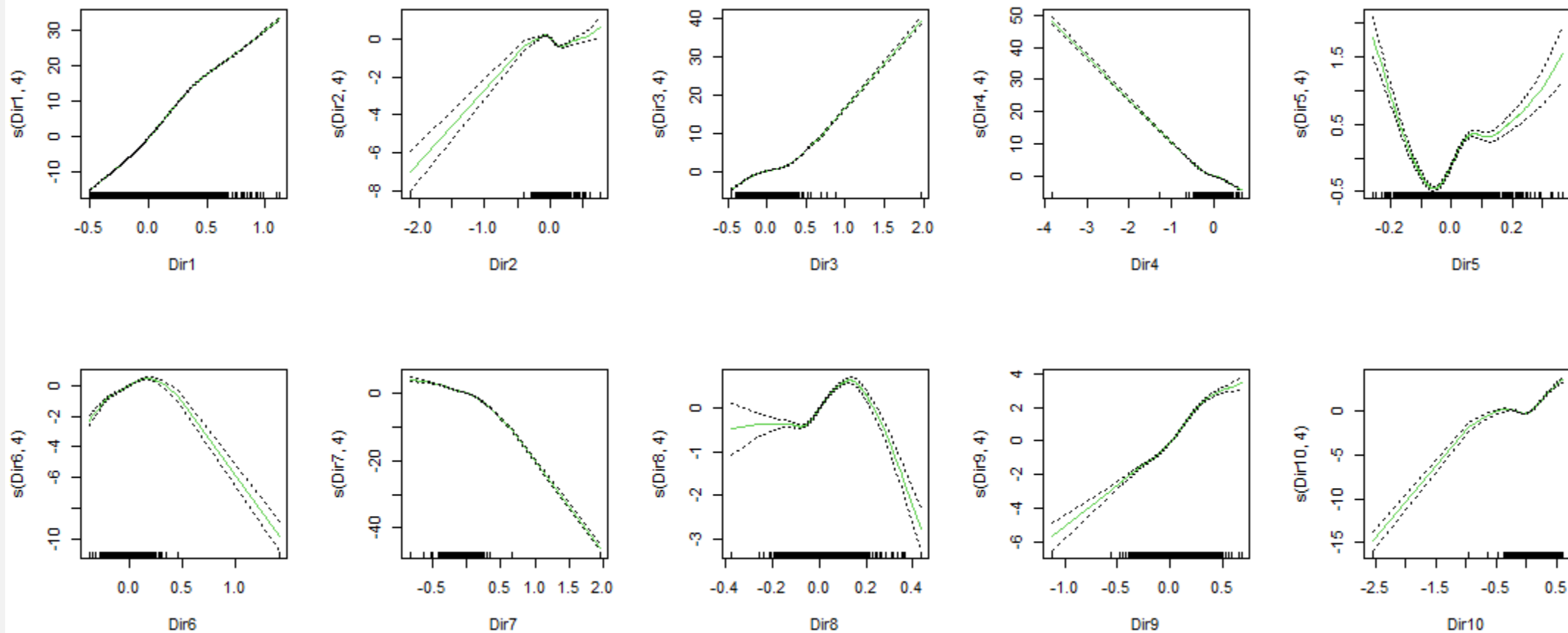
모델 개선 : GAM with 10 components

- 10개의 component를 사용한 GAM의 성능이 가장 좋았으므로 해당 모델을 선택해 모델 개선 진행
- 전체 component의 basis function으로 **smoothing splines** 사용
effective df = 2~10에서 10-fold cross validation으로 parameter tuning
- 전체 component의 basis function으로 **natural cubic splines** 사용
df = 2~10에서 10-fold cross validation으로 parameter tuning
- 10-fold cross validation 결과,
effective df = 4의 smoothing splines을 사용한 GAM이
가장 성능이 좋은 것으로 확인
(10-fold CV RMSE = 351.8112)



2-2. SIR (1)

모델 개선 : GAM with smoothing splines ($df = 4$)



- training set에 모델을 적합 후 GAM plot을 참고해 변수마다 basis function의 df 조정

2-2. SIR (1)

최종적으로 선택된 GAM

| 설명변수 | Basis function | 설명변수 | Basis function |
|------|--------------------------------------|------|--------------------------------------|
| PC1 | smoothing splines (effective df = 4) | PC6 | smoothing splines (effective df = 4) |
| PC2 | smoothing splines (effective df = 2) | PC7 | smoothing splines (effective df = 4) |
| PC3 | smoothing splines (effective df = 4) | PC8 | smoothing splines (effective df = 2) |
| PC4 | smoothing splines (effective df = 3) | PC9 | smoothing splines (effective df = 2) |
| PC5 | smoothing splines (effective df = 2) | PC10 | smoothing splines (effective df = 4) |

- 9개의 component를 사용한 GML의 10-fold CV RMSE = 355.0469
 - 10개의 component를 사용하고 basis function을 모두 smoothing splines (effective df = 4)으로 선택한 GAM의 10-fold CV RMSE = 351.8112
- ➡ 최종적으로 선택된 GAM의 10-fold CV RMSE = **348.9979**

2-2. SIR (2)

SIR을 통해 얻은 eigenvector를 살펴본 결과,

대부분의 변수들에 대한 PC loading의 절댓값이 0.1보다 작다는 것을 확인

| component | PC1 | PC2 | PC3 | PC4 | PC5 |
|---------------------------|-----|-----|-----|-----|------|
| # of loading ≥ 0.1 | 10 | 12 | 22 | 11 | 15 |
| component | PC6 | PC7 | PC8 | PC9 | PC10 |
| # of loading ≥ 0.1 | 20 | 9 | 7 | 15 | 13 |

➔ 102개의 설명변수 중 β 를 구성하는데 큰 영향을 주는 변수만 사용할 수 있을까?

2-2. SIR (2)


SIR을 통한 변수선택

Step.1 : 앞서 진행한 것과 같이 SIR을 통해 차원축소

Step.2 : 첫 30개 eigenvector를 선택 후,

각 eigenvector에서 loading의 절댓값 상위 30개의 설명변수를 추출

| | PC_1 | PC_2 | PC_3 |
|-------|--------|--------|--------|
| X_1 | 0.5 | 0.3 | 0.4 |
| X_2 | 0.4 | 0.0 | 0.0 |
| X_3 | 0.0 | 0.0 | 0.2 |
| X_4 | 0.3 | 0.3 | 0.0 |
| X_5 | 0.0 | 0.2 | 0.7 |
| X_6 | 0.0 | 0.0 | 0.0 |



| | PC_1 | PC_2 | PC_3 |
|-------|--------|--------|--------|
| X_1 | X_1 | X_1 | X_1 |
| X_2 | X_2 | X_4 | X_3 |
| X_4 | X_4 | X_5 | X_5 |

2-2. SIR (2)

Step. 3 : Step.2에서 추출된 변수들에 대해서

각 eigenvector마다 추출된 설명변수 그룹에 포함되는 비율이 얼마인지 계산

Step.4 : 포함 비율이 0.5를 넘는 설명변수만 선택

| | PC_1 | PC_2 | PC_3 |
|-------|--------|--------|--------|
| X_1 | 0.5 | 0.3 | 0.4 |
| X_2 | 0.4 | 0.0 | 0.0 |
| X_3 | 0.0 | 0.0 | 0.2 |
| X_4 | 0.3 | 0.3 | 0.0 |
| X_5 | 0.0 | 0.2 | 0.7 |
| X_6 | 0.0 | 0.0 | 0.0 |



| | PC_1 | PC_2 | PC_3 |
|-------|--------|--------|--------|
| X_1 | X_1 | X_1 | X_1 |
| X_2 | | X_4 | X_3 |
| X_4 | | X_5 | X_5 |



X_1 : 포함 비율 $3/3 = 1$

X_2 : 포함 비율 $1/3 = 0.33$

X_3 : 포함 비율 $1/3 = 0.33$

X_4 : 포함 비율 $2/3 = 0.66$

X_5 : 포함 비율 $2/3 = 0.66$

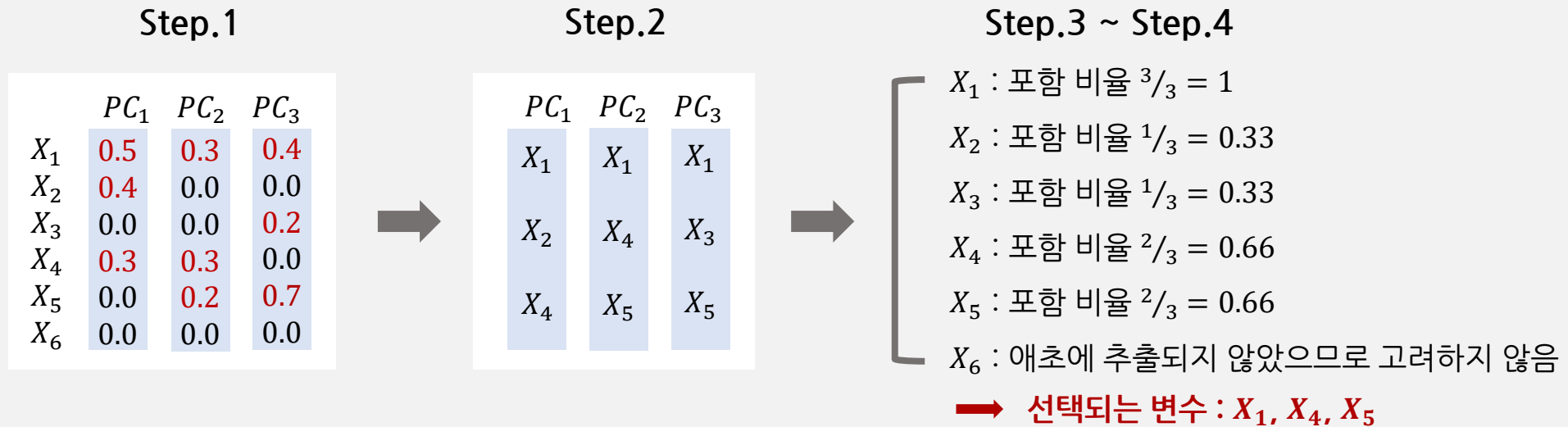
X_6 : 애초에 추출되지 않았으므로 고려하지 않음

➡ 선택되는 변수 : X_1, X_4, X_5

2-2. SIR (2)

SIR을 통한 변수선택 결과

- Step.1 → Step.2 에서 **75개의 설명변수가 추출**
- Step.3 → Step.4 에서 최종적으로 **27개의 설명변수가 선택**

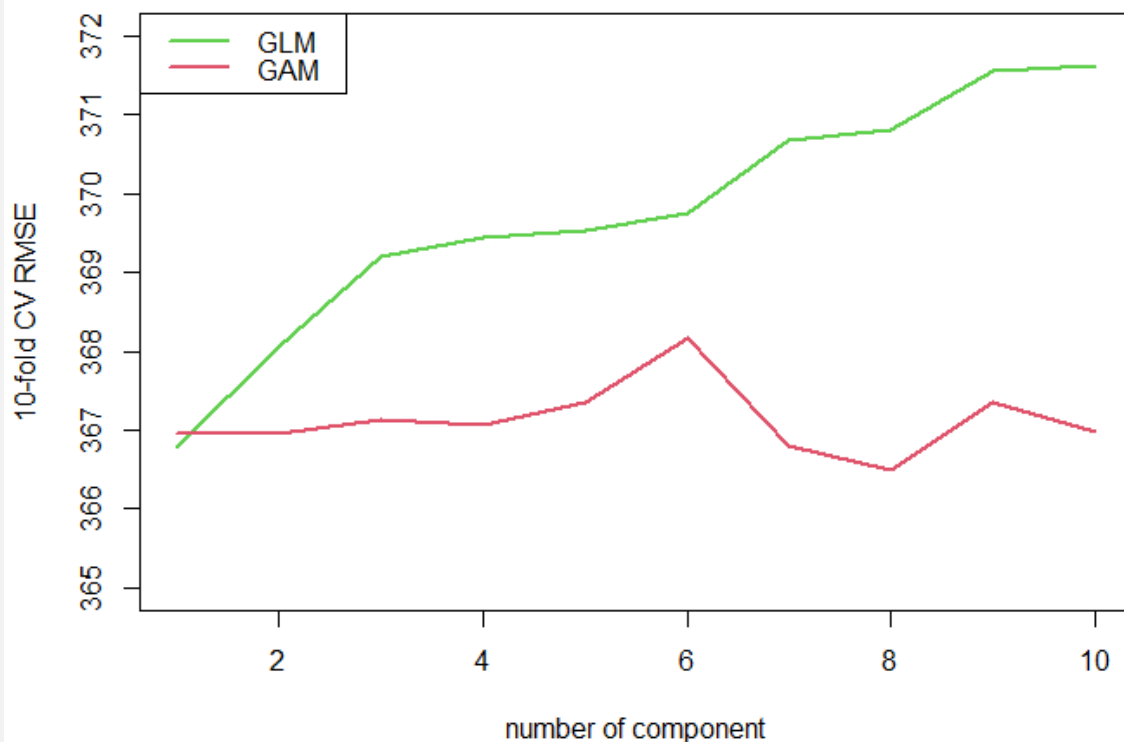


- 102개의 설명변수 → 최종적으로 선택된 27개의 설명변수를 사용해 **2차 SIR**을 진행

2-2. SIR (2)

2차 SIR 차원축소 진행

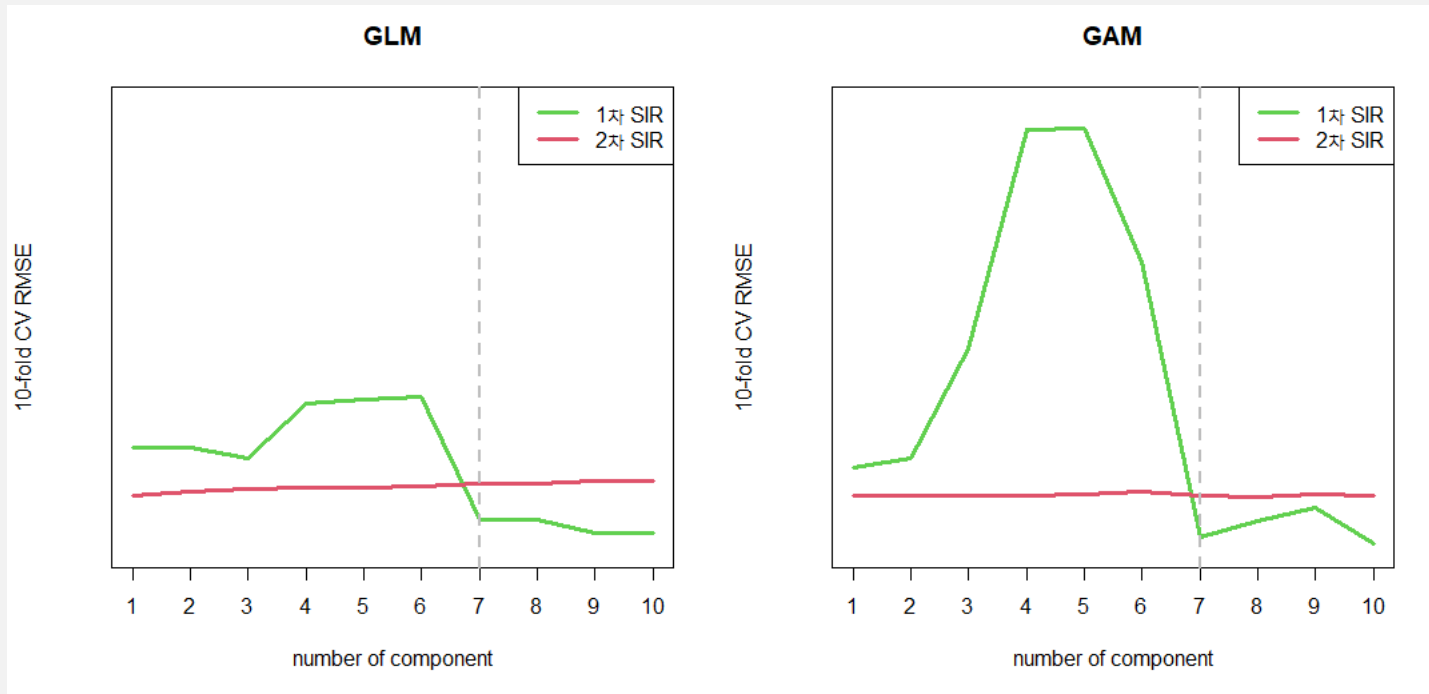
- 1차 SIR과 같이 1~10개의 eigenvector를 선택해 10-fold cross validation error 확인
- 사용한 모델 : GLM, GAM(각 변수의 basis function은 effective df=4인 smoothing spline)



- GLM은 1개의 component를 사용했을 때
CV RMSE = 366.7877로 가장 성능이 좋음
- GAM은 8개의 component를 사용했을 때
CV RMSE = 366.486로 가장 성능이 좋음
- component의 수가 많아질수록
GLM의 성능이 떨어지는 경향

2-2. SIR (2)

1차 SIR vs 2차 SIR 비교



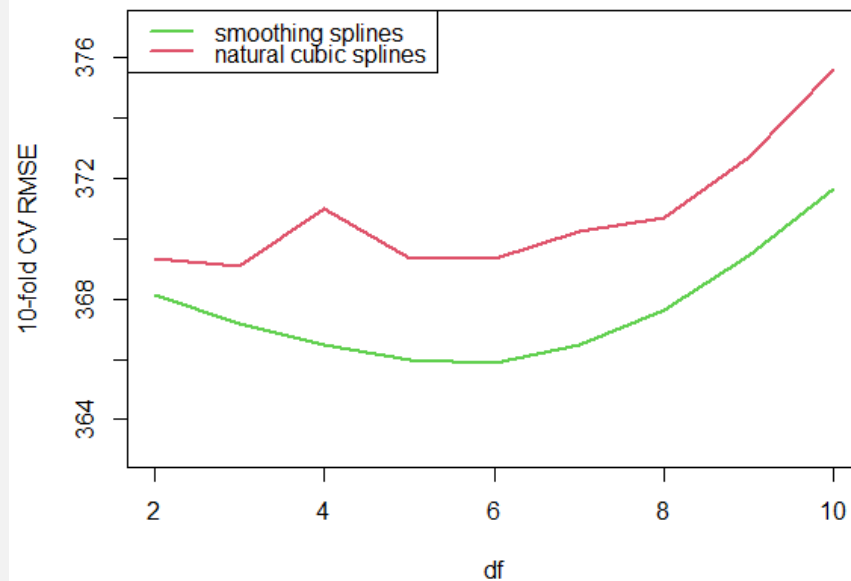
- GLM과 GAM 모두 component의 수가 **6개 이하일 때는 2차 SIR의 성능이 더 좋음**
- 이 경우 GLM보다 GAM의 성능 차이가 큼
- GLM과 GAM 모두 component의 수가 **7개 이상일 때는 1차 SIR의 성능이 더 좋음**
- 이 경우 GAM보다 GLM의 성능 차이가 큼

- 2차 SIR에서 변수 선택으로 추가적 정보 손실이 일어났음에도
7개 이상의 component를 사용한 GAM은 1차 SIR에서의 모델과 성능 차이가 크게 나지 않음
➡ 2차 SIR의 GAM에서 가장 성능이 좋았던 8개의 component를 사용한 GAM을 선택해 모델 개선 진행

2-2. SIR (2)

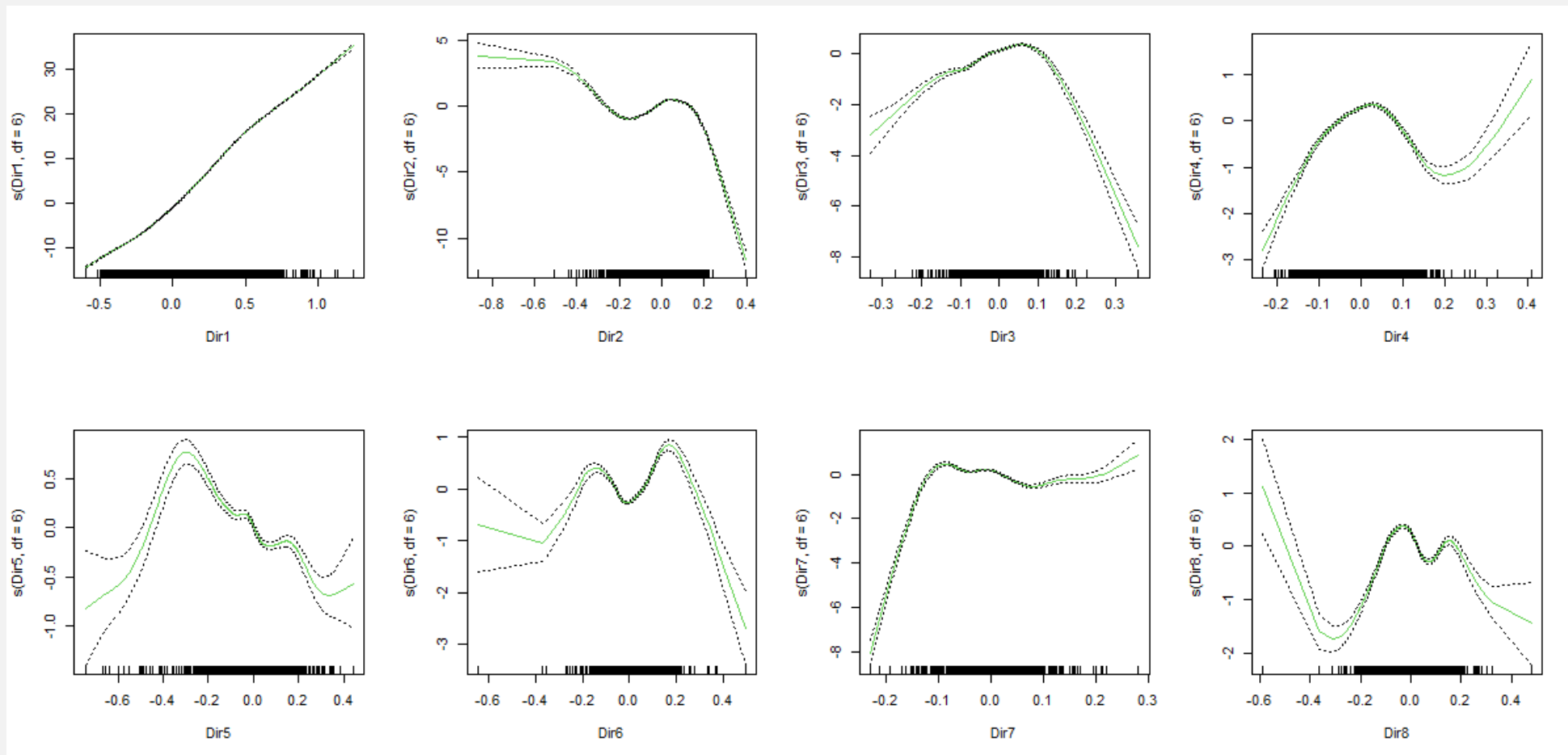
모델 개선 : GAM with 8 components

- 8개의 component를 사용한 GAM의 성능이 가장 좋았으므로 해당 모델을 선택해 모델 개선 진행
- 전체 component의 basis function으로 **smoothing splines** 사용
effective df = 2~10에서 10-fold cross validation으로 parameter tuning
- 전체 component의 basis function으로 **natural cubic splines** 사용
df = 2~10에서 10-fold cross validation으로 parameter tuning
- 10-fold cross validation 결과,
effective df = 6의 smoothing splines을 사용한 GAM이
10-fold RMSE = 365.9016로 가장 성능이 좋은 것으로 확인



2-2. SIR (2)

모델 개선 : GAM with smoothing splines (df = 6)



- training set에 모델을 적합 후 GAM plot을 참고해 변수마다 basis function 및 df 조정

2-2. SIR (2)

최종적으로 선택된 GAM

| 설명변수 | Basis function | 설명변수 | Basis function |
|------|--------------------------------------|------|--------------------------------------|
| PC1 | smoothing splines (effective df = 4) | PC5 | natural cubic splines (df = 4) |
| PC2 | smoothing splines (effective df = 6) | PC5 | natural cubic splines (df = 4) |
| PC3 | smoothing splines (effective df = 6) | PC7 | smoothing splines (effective df = 8) |
| PC4 | smoothing splines (effective df = 3) | PC8 | natural cubic splines (df = 5) |

- 1개의 component를 사용한 GML의 10-fold CV RMSE = 366.7877
- 8개의 component를 사용하고 basis function을 모두 smoothing splines (effective df = 4)으로 선택한 GAM의 10-fold CV RMSE = 366.486
- ➡ 최종적으로 선택된 GAM의 10-fold CV RMSE = **362.7237**
- 1차 SIR에서 최종 선택된 GAM (10-fold RMSE = 348.9979)보다는 성능이 떨어짐

3-1. 예측 성능 확인 및 비교

분석 과정에서 선택된 3개의 모델

- LASSO GAM, 1차 SIR GAM, 2차 SIR GAM

예측 성능 확인 및 비교

- Test set을 통해 test RMSE 확인 및 비교
- 범주율을 총 4단계로 나눠 범주 위험 단계 예측 성능 확인 및 비교

test set의 전처리

- training set의 설명변수 별 평균 및 표준편차를 사용해 test set의 설명변수 정규화
- SIR 방법의 경우 training set에서 얻었던 eigen vector를 사용

3-1. 예측 성능 확인 및 비교

test RMSE 확인 및 비교

| 모델 | 10-fold CV RMSE | test RMSE |
|-------------------|-----------------|-----------------|
| LASSO GAM | 356.2700 | 362.6672 |
| 1차 SIR GAM | 348.9979 | 326.2085 |
| 2차 SIR GAM | 362.7237 | 372.2866 |

- 세 모델 중 10-fold CV RMSE가 가장 작았던 '1차 SIR GAM'이 test RMSE도 가장 작음
- LASSO GAM과 2차 SIR GAM은 test RMSE가 10-fold CV RMSE보다 큰 반면
1차 SIR GAM은 오히려 test RMSE가 10-fold CV RMSE보다 작음
- **1차 SIR GAM의 예측 성능이 가장 뛰어남을 확인**

3-1. 예측 성능 확인 및 비교

범죄 위험 단계 예측 성능 확인 및 비교

- 범죄 예측 모델은 각 지역의 범죄율을 '정확히' 예측할 필요는 없음
- 범죄 예측 모델의 핵심은 **각 지역이 '얼마나 위험한지' 예측**하는 것

범죄 위험 단계 기준

- training set 반응변수의 분위수(25%, 50%, 75%)를 범죄 단계의 기준으로 설정

| 10만명 당 강력범죄 건수 (Y) | $Y < 161$ | $161 \leq Y < 374$ | $374 \leq Y < 783$ | $783 \leq Y$ |
|-----------------------|-----------|--------------------|--------------------|--------------|
| 범죄 위험 단계 | 1 (매우 안전) | 2 (안전) | 3 (위험) | 4 (매우 위험) |

- 위의 기준을 적용해 각 모델의 범죄 위험 단계에 대한 예측 성능 확인 및 비교

3-1. 예측 성능 확인 및 비교

범죄 위험 단계 예측 성능 확인 및 비교

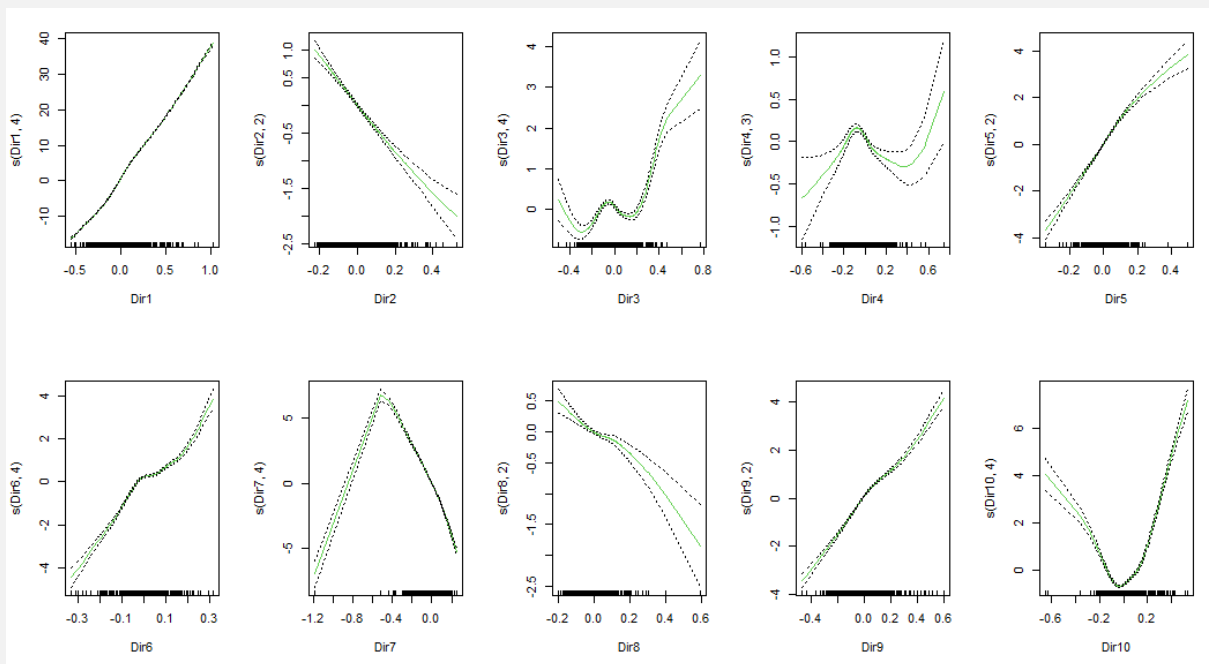
| | 예측 위험 단계 | | | | | | | | | | | | | | | |
|-------------|-----------|----|-----|-----|-----|------------|----|-----|-----|-----|------------|----|-----|-----|-----|-------|
| 실제 위험 단계 | LASSO GAM | | | | | 1차 SIR GAM | | | | | 2차 SIR GAM | | | | | total |
| | | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | |
| | 1 | 57 | 57 | 5 | 0 | 1 | 57 | 54 | 8 | 0 | 1 | 53 | 60 | 6 | 0 | 119 |
| | 2 | 19 | 75 | 37 | 5 | 2 | 21 | 75 | 35 | 5 | 2 | 20 | 71 | 36 | 9 | 136 |
| | 3 | 1 | 17 | 63 | 31 | 3 | 1 | 18 | 66 | 27 | 3 | 1 | 24 | 61 | 26 | 112 |
| | 4 | 0 | 6 | 36 | 91 | 4 | 0 | 2 | 37 | 94 | 4 | 1 | 4 | 39 | 89 | 133 |
| total | | 77 | 155 | 141 | 127 | | 79 | 149 | 146 | 126 | | 75 | 159 | 142 | 124 | 400 |

- LASSO GAM과 1차 SIR GAM은 실제 4단계를 1단계로 / 실제 1단계를 4단계로 예측하는 경우는 없음
- **2차 SIR GAM**은 실제 1단계를 4단계로 예측한 경우는 없지만 **실제 4단계를 1단계로 예측한 경우가 하나 존재**
- **4단계에 대한 예측 성능은 1차 SIR GAM이 가장 좋음** (133개의 4단계 지역 중 94개를 4단계로 분류)
- 세 모델 모두 실제보다 1, 4단계는 적게 예측하고 2, 3단계는 많이 예측하는 경향

3-2. 모델 해석

1차 SIR GAM과 2차 SIR GAM의 해석

- training set에 적합 후 GAM plot을 통해 각 component와 반응변수의 관계를 파악할 수 있음
- 하지만 각 component가 무엇을 의미하는지 해석하기 어려움

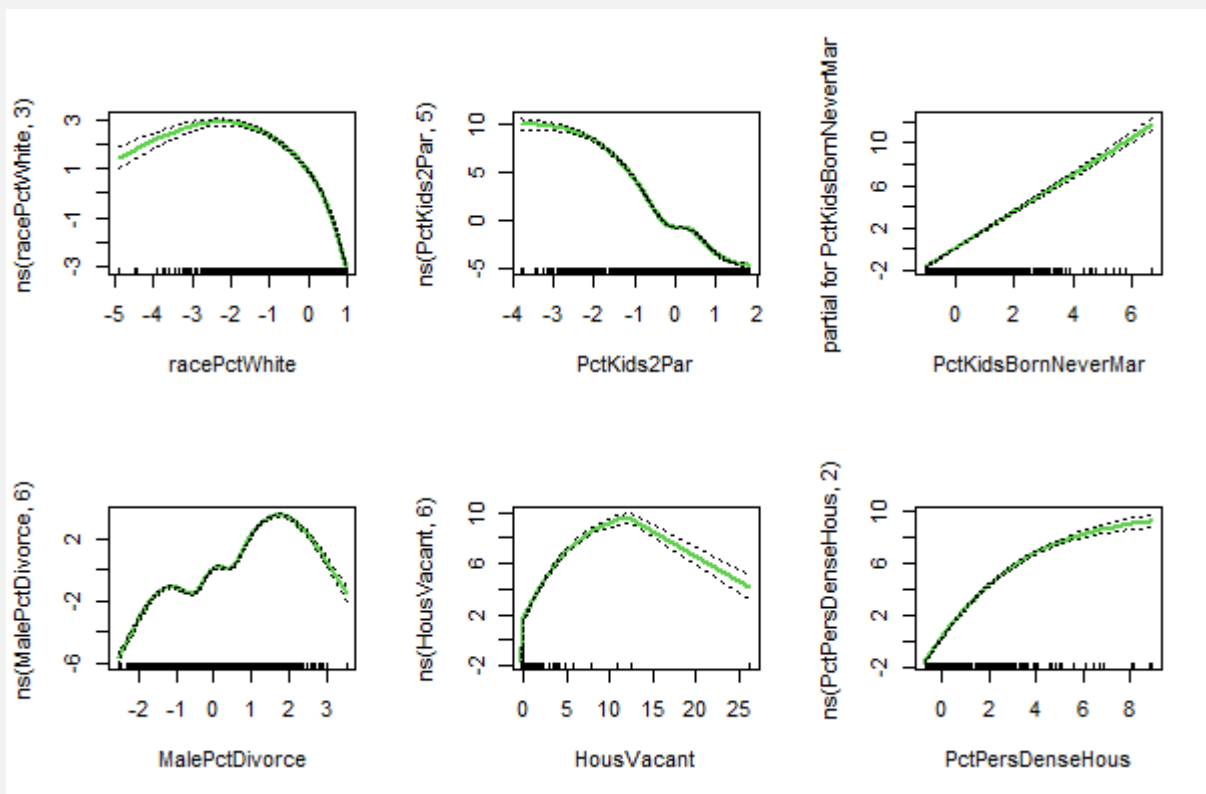


PC1 = 0.54(인구 수) - 0.52(도시 거주 인구 수) - 0.22(전체 이혼율) - 0.23(양부모 가정 아동 비율)
- 0.30(가구 소유자의 거주율) + 0.27(가구주의 거주율)

3-2. 모델 해석

LASSO GAM의 해석

- 설명변수가 명확하기 때문에 GAM plot을 통해 해석이 쉬움



- 백인비율이 낮으면 강력범죄율이 높으며, 백인비율이 평균에 가까워질수록 강력범죄율은 급격히 감소한다.
- 양부모 가정 아동비율이 높을수록 강력범죄율은 감소한다.
- 미혼 가정 아동비율이 높을수록 강력범죄율은 증가한다.
- 남성 이혼율이 높을수록 강력범죄율은 증가한다.
- 빈 집의 수가 많을수록 강력범죄율은 증가한다.
- 거주 인구 대비 좁은 집의 비율이 높을수록 강력범죄율은 증가한다.

3-3. 분석의 결론, 의의, 한계

분석의 결론

- 여러 설명변수를 사용한 범죄 예측 모델로 3가지 모델을 선택
: LASSO GAM, 1차 SIR GAM, 2차 SIR GAM
- 예측 성능은 1차 SIR GAM이 가장 좋음
- 모델의 해석은 LASSO GAM이 간단하고 이해하기 쉬움
- LASSO GAM을 통해서 강력범죄율에 영향을 미치는 설명변수를 파악하고
각 설명변수가 강력범죄율에 어떤 영향을 미치는지 파악 가능
- 1차 SIR GAM을 사용해 각 지역의 강력범죄율 예측 가능

3-3. 분석의 결론, 의의, 한계

분석의 의의

- 102개의 설명변수를 가진 데이터에서 변수 선택 및 차원 축소 방법을 적용해 5~10개의 설명변수만 사용되는 간단한 모델을 세움
- LASSO GLM에서 추가적인 변수선택과 GAM을 통해 간단하고 성능도 향상된 모델을 세움
- SIR 방법을 차원축소 뿐만 아니라 변수선택에도 사용

분석의 한계

- 이상치 제거를 하지 못함
- GAM plot에만 의존해서 각 변수의 basis function과 df를 결정
- SIR을 통한 변수선택을 진행해 2차 SIR GAM을 만들었지만 모델의 성능이 좋지 않음
- SIR 방법을 사용한 모델은 해석이 어려움

감사합니다