

Project for P-spline and Multilevel

Choi TaeYoung

2020-10-14

Contents

1	필요한 패키지	1
2	데이터	2
3	Multilevel 모델에 적용	3
3.1	Naive's GCV vector 찾기	4
4	그래프	4

1 필요한 패키지

2 데이터

- Y data : Y데이터의 경우 134주 (2018년 1월 ~ 2020년 7월) 동안의 카카오로 “혼자여행”을 검색한 횟수를 지역별로 나타냄
- X data : X데이터의 경우 17개의 지역별 인구수

3 Multilevel 모델에 적용

- 논문의 방법인 EM알고리즘을 통해 multilevel spline 방법으로 최적의 μ 벡터를 찾았다.

```
#multilevel

#beta_hat_vector 구하기

grain_out <- NULL
J=135
beta_hat <- NULL
for(m in 1:135){
  result2_out <- NULL
  results2 <- glm(unlist(y_list[m]) ~ unlist(x_list), maxit=2000)
  kth_beta_hat <- coef(results2)[2]
  kth_var <- diag(vcov(results2))[2]
  grain_out <- list(kth_beta_hat, kth_var)
  grain_out
  beta_hat <- rbind(beta_hat, grain_out)
}
```

- p-spline 기법을 활용하여 새롭게 짠 코드로 리얼데이터에 적용

```
lambda <- c(1e-03, 1e-02, 1, 10)
GCV_vec <- NULL

for(i in 1:length(lambda)){
  EM_out <- MultiEM_ps(x=z_month, beta_hat_vec = unlist(beta_hat[,1]), V = diag(unlist(beta_hat[,2])), lambda=
  GCV_vec <- rbind(GCV_vec, EM_out$GCV)
}
lambda[which.min(GCV_vec)]
```

```
## [1] 10
```

- lambda[which.min(GCV_vec)]을 실행할 때, 1000이 나온다.
- 그래서 1000근처에서 GCV벡터를 더 찾아보기로 한다.
- 최적의 GCV_vec로 EM_out구하기 <- mu_hat 구함

```
EM_out <- MultiEM_ps(x=z_month,
  beta_hat_vec=unlist(beta_hat[,1]),
  V=diag(unlist(beta_hat[,2])),
  lambda=1)

tail(EM_out$mu)
```

```
##           [,1]
## [130,] 0.016357017
## [131,] 0.017329484
## [132,] 0.017760590
## [133,] 0.017389418
## [134,] 0.015683051
## [135,] 0.009880741
```

- Multilevel과 성능을 비교하기위해서 Naive한 방법으로 구해보자.
- Naive기법 역시 P-spline으로 코드를 짰 후 실행했다.

```
#naive
GCV_vec <- NULL
lambda <- c(0.0001,0.001,0.01,0.1,1,10,100,1000, 10^4, 10^5)
for(i in 1:length(lambda)){
  naive_out <- naive_ps(x=z_month,
                        beta_hat_vec=unlist(beta_hat[,1]),
                        lambda[i])
  GCV_vec <- rbind(GCV_vec,naive_out$GCV)
}

lambda[which.min(GCV_vec)]
```

```
## [1] 10000
```

3.1 Naive's GCV vector 찾기

- Naive 역시 비슷한 방법으로 풀어나간다.

```
lambda[which.min(GCV_vec)]
```

```
## [1] 10000
```

```
naive_out <- naive_ps(x=z_month,beta_hat_vec=unlist(beta_hat[,1]),lambda = lambda[which.min(GCV_vec)])
```

```
tail(naive_out$mu)
```

```
##           [,1]
## [130,] 0.01149707
## [131,] 0.01126898
## [132,] 0.01104293
## [133,] 0.01081752
## [134,] 0.01057299
## [135,] 0.01013894
```

4 그래프

```
# hat_all
single_beta <- unlist(beta_hat[,1]) %>% as.vector()
mu_z_multi <- EM_out$mu %>% as.vector()

test_mon <- fread("soltr_kakao_y.csv")
test_mon <- test_mon[-1,1]

mu_z <- cbind(test_mon,mu_z_multi) %>% as.data.frame
mu_z <- rename(mu_z, Week = V1)

df2 <- cbind(test_mon,single_beta)
df2 <- rename(df2, Week =V1)

df2_naive <- naive_out$betaEst
df2 <- cbind(df2,df2_naive)
```

```

df2 <- rename(df2, Naive =df2_naive)

df2 <- cbind(df2,mu_z$mu_z_multi)
df2 <- rename(df2, Multi =V2)

# gather함수 사용
df2 <- gather(df2[, c("Week", "single_beta", "Naive", "Multi")],
              key = "Method", value = "mu_z", -Week)
df2$Week <- parse_date_time(df2$Week, "ymd")
df2$Week <- as.Date(df2$Week, format="%Y-%m-%d")

# 그래프 그리기
g <- ggplot(df2, aes(x=Week, y=mu_z, group=Method)) +
  geom_line(data= df2 %>% dplyr::filter(Method != "single_beta"),aes(x = Week, y = mu_z, color = Method)) +
  geom_point(data=df2 %>% dplyr::filter(Method == "single_beta"), aes(x = Week, y = single_beta, color = Method)) +
  guides(linetype = "none") +
  scale_color_discrete(name = "Method")
g

```

