Application of emIRT

Reference Paper

1. Fast Estimation of Ideal Point with Massive Data, KOSUKE IMAI et al.
2. R package: emIRT

이번 리뷰에서는 저번에 살펴본 Fast Estimation of Ideal Point with Massive Data 논문의 알고리즘을 R에서 수행해보며 결과를 살펴보려고 한다. 미국의 선거 데이터, 일본의 설문조사 데이터, 그리고 가능하다면 한국의 데이터까지 적용한 결과를 담아보겠다.

Brief Review of Ideal Point Estimation

N명의 입법자가 있고 이들이 J개의 법안에 찬성/반대 여부를 기록한 데이터가 있다고 가정하자. 가령 보수에 가까운 입법자들은 그들의 성향을 따르는 법안에 찬성 표를 던질 것이다. 이렇게, 그들의 성향이 투표에 드러나고 바로 이를 EM을 통해서 추정하는 것이 Ideal Point Estimation이다. 본 논문에서 EM이 가지는 큰 장점은 바로 빠르다는 것이다. 기존에도 이와 유사한 주제로 Bayesian의 계층 모형 등이 사용 되었는데, MCMC에 기반한 베이지안 모형은 데이터의 크기가 클수록 속도가 현격히 낮아진다. 이와 반대로, 본 논문에서 제시하는 EM은 속도가 훨씬 빠르다. 더불어, 일부 경우를 제외하고 추정량을 closed-form으로 유도하기 때문에 정확도 측면에서 MCMC와 다를 것이 없다. 즉, 결과는 달라지지 않으면서 속도는 현격히 빨라지는 것이다.
본 논문에서는 여러 데이터의 형태에 따른 EM을 제시한다. 첫째는 y가 binary일 때, 둘째는 y가 세 개의 클래스를 가지며 ordinal일 때, 셋째는 시간 변수가 추가된 dynamic model, 넷째는 계층 모형, 다섯째는 텍스트 데이터, 여섯째는 네트워크 데이터에 대한 분석이다.

Overview of Data

Asahi-Todai Elite Survey (ordinal model에 사용됨)
Asahi Shimbun 신문사와 도쿄 대학의 협업으로 이루어진 설문조사지이다. 데이터를 우선 불러와보자.

```
library(emIRT)
library(ggplot2)
library(glue)

data("AsahiTodai")
data = AsahiTodai
dim(data$dat.all)

## [1] 19443    98

head(data$dat.all)[,1:9]
```

```
##      defense nuclear1 treaty attack un smallgov lifetime publicen keynes
## [1,]       1        1      2      3  1        2        3        2      2
## [2,]       3        1      3      3  3        3        1        2      3
## [3,]       1        3      1      1  1        1        2        1      2
## [4,]       1        1      1      1  1        2        1        1      3
## [5,]       1        1      2      1  2        3        1        1      2
## [6,]       1        3      2      2  1        2        1        3      1
```

총 19443개의 행과 98개의 열로 이루어져 있다. 즉, 설문조사에 응한 사람들은 19443명이고 설문지의 질문은 98개이다. 98개 중, 10개만 출력을 했는데, defense, nuclear에 대해서 1부터 3까지의 값으로 표시가 되어 있다. 원래 응답의 scale은 다섯 단계였지만 클래스가 세 개일때 EM의 closed form이 유도되므로 클래스를 3개로 압축했다.

109th US Senate Data(binary model에 사용됨)
109번째 미국 상원들이 어떤 법안에 찬성/반대표를 던졌는지에 대한 데이터이다. 데이터를 살펴보자.

```
data(s109)
head(s109$votes)[,1:10]
```

```
##                   1-1 1-2 1-3 1-4 1-5 1-6 1-7 1-8 1-9 1-10
## BUSH (R USA)        9   1   1   9   9   9   9   9   1    1
## SESSIONS (R AL)     6   1   1   1   1   6   6   6   1    1
## SHELBY (R AL)       9   1   1   1   1   6   6   6   1    1
## MURKOWSKI (R AK)    9   1   1   9   1   6   6   6   1    1
## STEVENS (R AK)      6   1   1   1   1   6   6   6   1    1
## KYL (R AZ)          9   1   1   1   1   6   6   6   1    1
```

예를 들어 BUSH 의원은 1-1 법안에 9(중립)을, 1-2 법안에 1(긍정) 표를 던졌다. 또 SESSIONS 의원은 1-1 법안에 6(부정) 표를 던졌다.

Martin-Quinn Judicial Ideology Scores(dynamic model에 사용됨)
이 데이터는 1937년 10월부터 2013년 10월까지 총 77개의 session에 걸친 입법자들의 투표 결과를 기록한 데이터이다. 시간 변수가 있으므로 dynamic ideal point model에 사용되며, 논문의 notation인 $\overline{T}_i, \underline{T}_i$에 대한 정보가 있다. 즉, 입법자들이 언제 국회에 들어왔는지, 그리고 언제 나갔는지에 대한 정보가 있다. 다른 데이터와 유사하게 start values과 priors가 있다. 데이터를 살펴보자.

```r
data(mq_data)
names(mq_data$data.mq)
```

```
## [1] "rc"           "startlegis"   "endlegis"     "bill.session" "T"
```

```r
# vote result for 45 legislators, about 5164 bills
dim(mq_data$data.mq$rc)
```

```
## [1]   45 5164
```

```r
head(mq_data$data.mq$rc[,1:10])
```

```
##          1 2 3 4 5  6  7  8  9 10
## Harlan   0 0 0 0 0  0  0  0  0  0
## Black    1 1 1 1 1 -1 -1 -1 -1  1
## Douglas  0 0 0 0 0  0  0  0  0  0
## Stewart  0 0 0 0 0  0  0  0  0  0
## Marshall 0 0 0 0 0  0  0  0  0  0
## Brennan  0 0 0 0 0  0  0  0  0  0
```

```r
# startlegis stores the starting point for each legislators
head(mq_data$data.mq$startlegis)
```

```
##      [,1]
## [1,]   17
## [2,]    0
## [3,]    1
## [4,]   21
## [5,]   30
## [6,]   19
```

```r
# endlegis stores the end point for each legislators
head(mq_data$data.mq$endlegis)
```

```
##      [,1]
## [1,]   33
```

```
## [2,]   33
## [3,]   37
## [4,]   43
## [5,]   53
## [6,]   52
```

```
# there are 77 sessions in total (1937 ~ 2013)
table(mq_data$data.mq$bill.session)
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  48  45  41  44  58  77  80  92  73  82  76  85  57  60  64  84  54  50  54  86
##  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
##  90  68  86  87  65  77  73  61  58  75  75  70  69  80  90 108 100  86  89  94
##  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59
##  88  83  99  82  95  90  83  79 101 105  78  81  87  68  66  57  52  46  42  41
##  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76
##  43  45  45  44  48  43  41  47  34  44  44  50  50  43  43  39  27
```

예를 들어, black이라는 의원을 살펴보자. 이 의원은 startlegis 값이 0으로 보아, 1937년 session에 참여하기 시작했고 endlegis 값이 33인 것으로 보아, 33년 후에 session에 참여하지 않은 것으로 보인다. 데이터를 살펴보면, 이 의원은 첫 번째 법안에 찬성 등을 한 것으로 파악된다.

dwnom data(Hierarchical model에 사용됨)
dwnom data는 입법가의 소속된 정당과 시간에 따른 법률의 찬/반 여부의 정보를 담은 데이터이다. 데이터는 아래의 값이 list의 형태로 저장되어 있다.

- y: $L \times 1$ 벡터로, 찬성, 반대의 1, -1 값을 가짐.
- i: $L \times 1$ 벡터로, 각 투표를 누가 했는지 나타내는 index. 즉, 이 데이터에서 의원이 총 3162명이 등장하는데, 이 3162명 중 누가 투표를 했는지 0에서 3161까지의 값으로 표시되어 있음.
- j: $L \times 1$ 벡터로, 법률에 대한 index. 즉, 이 데이터는, 총 20246 종류의 법안이 있으며 인덱스가 0부터 20245의 값을 가짐.
- g: $I \times 1$ 벡터로, 각 입법가의 group membership indicator이다. 여기서는 총 3162명의 의원이 있으니까 벡터의 차원이 3162×1이며, 총 526개의 정당이 있다. 즉, index는 0부터 526의 값을 가진다.

- z: $I \times D$ numeric의 관찰된 covariates의 행렬이다. 행은 입법가들에, 열은 D개의 covariates에 해당된다. 첫 번째 열은 절편으로 모두 1로 고정되며 다른 칼럼은 session과 같은 covariates이 될 수 있다. 만약 session으로 covariates이 된다면, time 변수가 들어가는 것이므로 dynamic ideal point model의 결과와 유사해질 것이다.

Useful Functions

emIRT 패키지에서는 아래와 같은 유용한 함수를 제공한다.

- convertRC: RollCall Matrix의 형태가 s109와 같이 1,6,9의 형태로 되어있을 경우, 이를 -1,0,1의 부정, 결측, 긍정으로 바꿔준다. 논문에서는 결측 값이 missing at random이라고 가정하고 논의를 진행한다.

- makePriors: makePriors(N, J, D)
  binIRT 함수를 위한 diffuse priors을 만들어준다. 논문의 standard ideal point model을 보면, $\mathbf{x}_i, \tilde{\boldsymbol{\beta}}_j$에 k-variate Normal random variable의 conjugate prior을 부여한다고 나와 있다. 여기서 $\mathbf{x}_i \sim N_k(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})$, $\tilde{\boldsymbol{\beta}}_j \sim N_{k+1}(\boldsymbol{\mu}_{\tilde{\boldsymbol{\beta}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{\beta}}})$ 인데, 각 모수의 hyper-parameter을 부여한다.
  - makePriors$x$mu: prior mean vectors for respondent ideal points x_i
  - makePriors$x$sigma: prior covariance matrix for respondent ideal points x_i
  - makePriors$beta$mu: prior mean vectors for α_j, β_j
  - makePriors$beta$sigma: prior Covariance matrix for α_j, β_j

- getStarts: getStarts(N, J, D)
  binIRT를 위한 initial value을 만들어준다.
  - alpha: $J \times 1$ initial vector for parameter α
  - beta: $J \times D$ initial matrix for parameter β
  - $\mathbf{x}$: $N \times D$ initial matrix for the respondent ideal points x_i

The Ideal Point Model

The Standard Ideal Point Model

이제 위 데이터로 standard ideal point model을 적합해본다. 우선 현재 개발된 software는 ideal point가 1차원인 경우만 다룬다. 즉, 각 입법자를 나타내는 ideal 점수가 vector가 아닌 points로 나온다.

```
rc <- convertRC(s109)
p <- makePriors(rc$n, rc$m, 1)
s <- getStarts(rc$n, rc$m, 1)
## Conduct estimates
```

```
lout <- binIRT(.rc = rc, .starts = s,  .priors = p,
.control = {list(threads = 1, verbose = FALSE, thresh = 1e-6)})


##
## ====================
## binIRT: Binary IRT via Expectation Maximization
##
##  Done in 71 iterations, using 1 threads.
## ====================
```

위 출력물 lout에는 아래의 결과들이 저장된다.

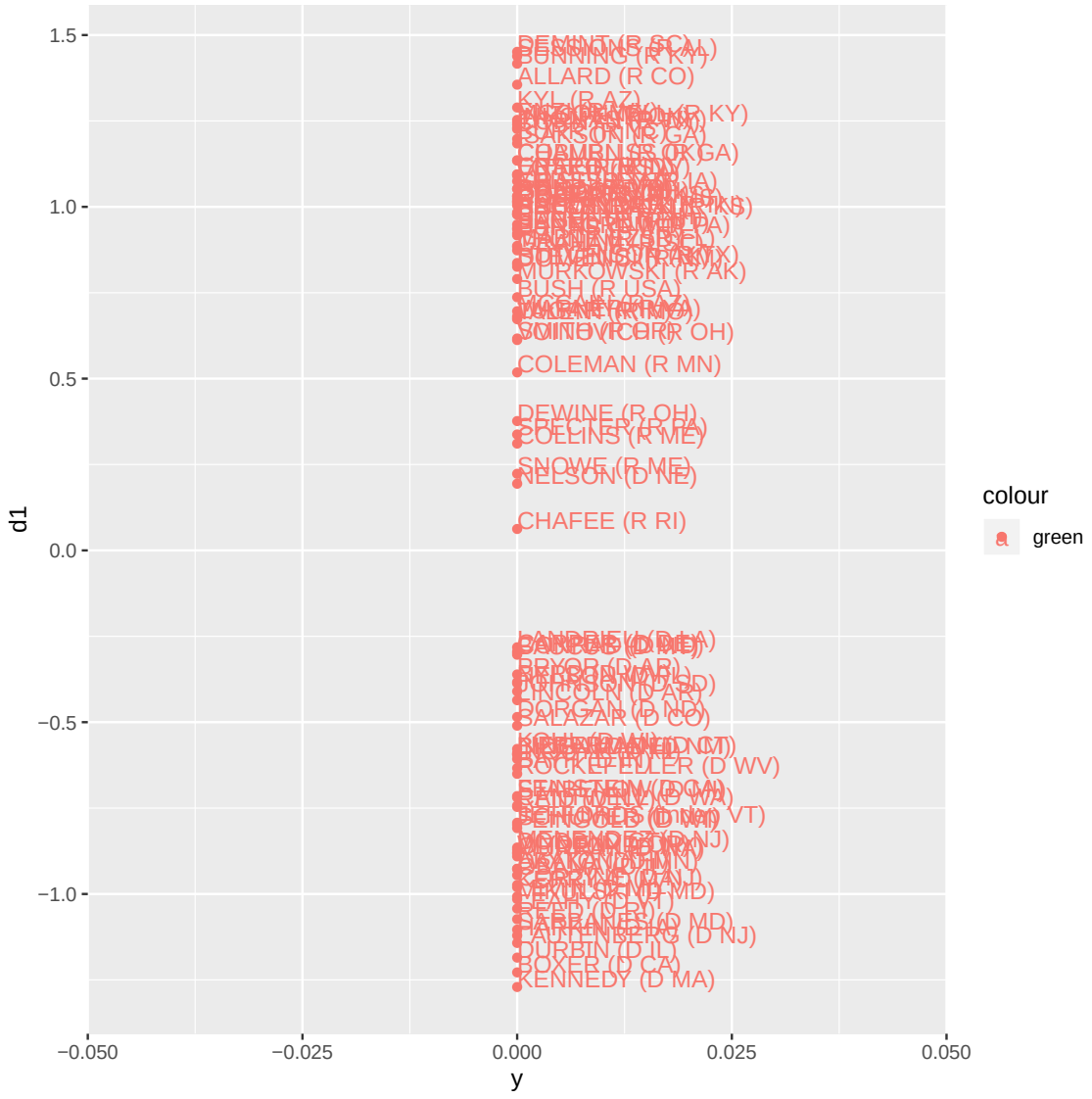
- lout$means$x: $N \times 1$ matrix of point estimates for the respondent ideal points x_i
- lout$means$vars: $J \times (D + 1)$ matrix of point estimates for the item parameters α, β

```
i_esti = lout$means$x
glue('ideal points가 최대인 의원은 {row.names(i_esti)[which.max(lout$means$x)]}입니다.')


## ideal points가 최대인 의원은 DEMINT (R SC)입니다.


glue('ideal points가 최소인 의원은 {row.names(i_esti)[which.min(lout$means$x)]}입니다.')


## ideal points가 최소인 의원은 KENNEDY (D MA)입니다.
```

추정된 ideal points 값을 보면, 최대인 의원은 DeMint이다. 위키 백과를 보면 DeMint is a member of the Republican Party and is aligned with the Tea Party movement. In 2011, DeMint was identified by Salon as one of the most conservative members of the Senate. 라고 나온다. 즉, 가장 보수적인 의원이라 ideal points 값도 가장 극단적으로 나온 듯하다. 그러면 보수의 대척점에 있는 진보 진영에는 Kennedy가 있는데 이는 꽤 앞뒤가 맞는 결과이다. In 2004 Kennedy endorsed Democratic presidential candidate John Kerry over George W. Bush. 라고 위키백과에 나와있다.
아래는 ggplot으로 시각화를 시도하였는데, 글자가 겹쳐서 자세히 보이지는 않는다. 아무튼, 이런 식으로 시각화를 한다면 진보와 보수 진영을 한번에 볼 수 있을 것이다.

```
i_esti = as.data.frame(i_esti)
i_esti$y = rep(0, nrow(i_esti))
```

```
ggplot(i_esti, aes(x= y, y= d1, colour="green", label=row.names(i_esti)))+
     geom_point() +geom_text(aes(label=row.names(i_esti)),hjust=0, vjust=0)
```



Estimate Variance of estimates using Parmetric Boostrap

이 논문에서는 parametric boostrap을 사용한 variance estimation을 제공하여 추정치의 불확실성에 대해 control 하고자 한다.

```
boot.bin <- boot_emIRT(lout, .data = rc, .starts = s, .priors = p,
.control = list(threads = 1, verbose = FALSE, thresh = 1e-06), Ntrials=10, verbose=2)


##
##   Iteration 2 complete...
##   Iteration 4 complete...
```

```
##   Iteration 6 complete...
##   Iteration 8 complete...
##   Iteration 10 complete...
```

```
boot.bin$bse$x[1:10]
```

```
##  [1] 0.07456719 0.04784480 0.05297747 0.05080933 0.05427243 0.06336177
##  [7] 0.04314188 0.04550447 0.03732242 0.07883147
```

위 결과는 미국 의원들의 ideal point estimates에 대한 parametric boostrap variance이다. 그럼 이 추정 분산을 이용하여 추정치 / standard error로 가설 검정을 할 수 있을까? (더 찾아보기)

The Model with a Three-category Ordinal Outcome

이제 세 개의 ordinal category을 가지는 데이터에 대해서 모델을 적합해본다. 논문에서 살펴보았듯이, closed-form으로 유도가 가능하기 때문에 MCMC의 추정치와 큰 차이가 없으며 오히려 속도는 훨씬 빠르다. 데이터는 Ashai 설문조사 데이터를 사용한다. 이 데이터는 원래 응답 스케일이 5였지만 ordinal IRT 적용을 위해 3개의 클래스로 만들었다는 것은 이미 앞서 언급했다.

```
lout_ord <- ordIRT(.rc = AsahiTodai$dat.all,
.starts = AsahiTodai$start.values,
.priors = AsahiTodai$priors, .D = 1,
.control = {list(verbose = TRUE,
thresh = 1e-6, maxit = 500)})
```

```
##
## ====================
## ordIRT: Ordinal IRT via Expectation Maximization
##
## Iteration: 50
## Iteration: 100
## Iteration: 150
##  Done in 170 iterations, using 1 threads.
## ====================
```

```
# see results for ideal points for voters
lout_ord$means$x[1:10]
```

```
##  [1] -0.1749753  1.7792414 -1.5114422 -1.0214353 -0.3737067 -0.4622271
##  [7] -0.8267817 -1.0303717  0.4800924 -0.9214892
```

binIRT와 유사하게 초기값과 prior을 지정해주어야 한다. 또한 결과 값으로 ordinal IRT의 모수 추정치를 반환한다. 각 추정치의 의미는 논문을 보면 자세히 알 수 있다.

- lout_ord$means$x: vector of points estimates for the respondent ideal points x_i
- lout_ord$means$beta: matrix of point estimates for the reparameterized item discrimination parameter β^*
- lout_ord$means$tau: matrix of point estimates for the bill cut point α^*
- lout_ord$means$Delta_sq: matrix of point estimates for the squared bill cut point difference τ_j^2

마찬가지로, $means$x로 1차원으로 추정된 ideal points를 확인할 수 있다.

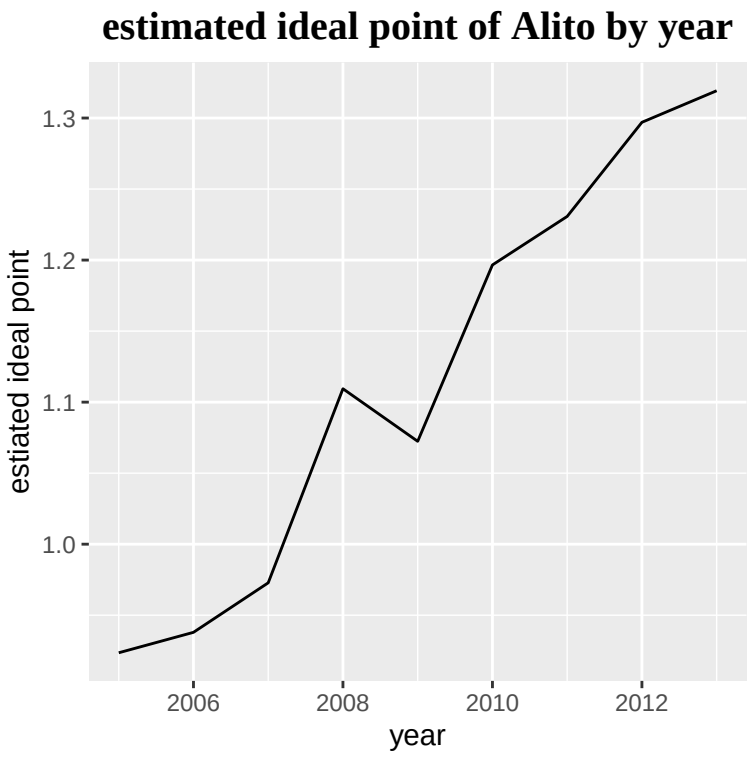
Dynamic Ideal Point Model

앞서 살펴본 mq_data로 dynamic ideal point model을 적합해본다.

```
lout_dyn <- dynIRT(.data = mq_data$data.mq,
.starts = mq_data$cur.mq,
.priors = mq_data$priors.mq,
.control = {list(threads = 1, verbose = TRUE, thresh = 1e-6, maxit=500)})


##
## ====================
## dynIRT: Dynamic IRT via Variational Inference
##
## Iteration: 50
##  Done in 51 iterations, using 1 threads.
## ====================


row_names = row.names(mq_data$data.mq$rc)
```

아래와 같이 특정 입법자의 연도에 따른 ideal point 변화 추이를 보여주는 함수를 작성해보았다.

```
plot_dynamic_ideal_point = function(dyn_model, row_names, legislator){
year = 1937:2013
name_indx = row_names[which(row_names==legislator)]
ideal_point_vec = dyn_model$means$x[name_indx,]
ideal_point_index = which(ideal_point_vec != 0)
x = year[ideal_point_index]
y = ideal_point_vec[ideal_point_index]
temp = data.frame(x,y); names(temp) = c('year', 'estimated_ideal_point')
ggplot(temp, aes(x=year, y=estimated_ideal_point)) +
geom_line() +      labs(x='year', y='estiated ideal point') +
ggtitle(glue('estimated ideal point of {legislator} by year')) +
theme(plot.title = element_text(family = "serif",
face = "bold", hjust = 0.5, size = 15, color = "black"))
}
plot_dynamic_ideal_point(dyn_model = lout_dyn, row_names = row_names, legislator = "Alito")
```



Aliot라는 사람의 ideal point는 시간이 지날수록 증가하는 것을 알 수 있다.
굳이 입법자의 법률에 대한 찬/반 여부가 아니더라도 시간에 따라서 기록된 어떤 범주형 변수에 대해서도, 이에 대한 latent score를 모델링하여 그 추이를 파악할 수도 있을 것 같다. 예를 들어서 어떤 사람의 달별 특정 매장 방문 여부에 대한 데이터가 있다고 할 때, 이에 대한 dynamic ideal point을 모델링한다면, 그 사람의 매장에 대한 선호 여부를 알 수 있을 것이다.

Hierarchical Ideal Point Model

마지막으로, 어떤 covariate의 함수로 ideal point을 추정하는 Hierarchical Ideal Point Model을 적합해보자.

```
data(dwnom)
lout_hier <- hierIRT(.data = dwnom$data.in,
.starts = dwnom$cur,
.priors = dwnom$priors,
.control = {list(threads = 8, verbose = TRUE,
thresh = 1e-4, maxit=200, checkfreq=1 )})

head(lout_hier$means$gamma)
head(lout_hier$means$x_implied)
```

hierIRT에서 눈여겨봐야할 결과는 아래와 같다.

- lout_hier$means$gamma: $I \times D$ 행렬로, group level의 계수 추정치인 $\hat{\gamma}_m$. 절편과 covariate의 추정치.
- lout_hier$means$x_implied: $I \times 1$ 벡터로, 각 입법가들의 추정된 ideal point. 이는 γ, z, η의 점 추정치의 함수로 게산된다.

우선, 위의 hierIRT을 돌리는데 거의 두 시간이 걸렸다. 다른 EM에 비해서 수렴하는데 많은 시간이 걸리는듯 하다. 결과를 아래와 같이 살펴보자.

```
> head(lout_hier$means$gamma)
            [,1]         [,2]
[1,] -0.19171152  0.017361451
[2,] -0.34256132  0.042019370
[3,] -0.20535229  0.014336297
[4,] -0.16043589  0.000000000
[5,]  0.03495931  0.008212859
[6,] -0.08064173 -0.001361449
```

예를 들어, 첫 번째 의원인 SPARKMAN의 ideal point을 절편과 time covariate으로 모델링한 계수는 위의 -0.19, 0.017이다. 논문에서 ideal point을 아래와 같이 모델링 했었다.

$$y_\ell^* = \alpha_{j[\ell]} + \beta_{j[\ell]} \boldsymbol{\gamma}_{g[j[\ell]]}^T \mathbf{z}_{i[\ell]} + \beta_{j[\ell]} \eta_{i[\ell]} + \epsilon_\ell$$

여기서 추정해야할 모수는 $\alpha, \beta, \gamma, \eta$이고 z는 covariate이다 (논문에서는 η가 random variable이긴 한데, emIRT를 보면 error term η에 대한 추정치의 output이 나와있다. 이 부분은 좀 더 생각을 해봐야 할 것 같다.) 여기서 첨자를 다시 정리해보면

- $j[\ell]$: roll call index, 이 데이터에서는 총 20246개의 roll call이 있음.
- $i[\ell]$: legislator index, 이 데이터에는 총 3126명의 legislators가 있음.
- $g[j[\ell]]$: group indicator index, 이 데이터에는 총 526개의 group(정당)이 있음.
- ℓ : votes index, 이 데이터에는 총 1879731개의 투표가 있음.

따라서 각 투표의 ideal point(latent score)인 y_ℓ^*을 roll call을 index로 가지는 $\alpha_{j[\ell]}, \beta_{j[\ell]}$와 group indicator을 index로 가지는 $\boldsymbol{\gamma}_{g[j[\ell]]}$와 legislator을 index로 가지는 $\mathbf{z}_{i[\ell]}, \eta_{i[\ell]}$로 모델링하는 것이다.

```
head(dwnom$data.in$i)
head(dwnom$data.in$y)
head(dwnom$data.in$j)
head(dwnom$data.in$g)
```

위 코드의 결과를 살펴보면 첫 번째 투표는 0 index을 가지는 입법자가 투표한 것으로 1의 값, 즉 찬성표를 던졌다. j 값을 살펴보면 이 법안은 3 index을 가지고 g 값을 살펴보면 0 index을 가진다. 즉 0 index을 가지는 입법자는 0 index의 group에 속해있다는 것이다. 이제 이 정치가의 ideal point을 모델링한 결과를 살펴보자. 이 의원은 SPARKMAN으로 1930년대에 활동했던 정치가이다. 이 정치가의 estimated ideal point가 time covariate과 가지는 관계는 결국 그 추정된 계수로, $\beta_{j[\ell]}\boldsymbol{\gamma}_{g[j[\ell]]}^T$에 대한 추정치일 것이다. 이는 각각 lout$means$beta[1]와 lout$means$gamma[1,][2]로, 3.05, 0.017이다. 이를 곱해보면 대략 0.05로, 미약하게나마 시간이 지날수록 SPARKMAN의 estimated ideal point는 증가함을 알 수 있다. 물론 이 계수가 유의한지는 parametric bootstrap으로 estimated variance을 구한다음에 검정(?)을 해봐야 할 것 같다.

Conclusion

이상으로 EM 활용하여, 다양한 형태의 데이터에 대한 latent score를 모델링하는 방법을 살펴보았다. 앞으로 더 살펴봐야할 점을 아래와 같이 정리해보았다.

- 굳이 정치 데이터가 아닌 latent score의 개념을 활용할 수 있는 어떤 데이터에도 이를 활용할 수 있다는 것이다. 추후에 다른 정치 데이터가 아닌, 다른 데이터로 분석을 하여 시각을 좀 더 넓혀야겠다.
- 통계학이 머신러닝과 가장 비교되는 점은, 분산을 통해 불확실성을 제어할 수 있다는 점이다. 해당 논문에서는 EM을 통해 추정치를 구했고 parametric boostrap을 통해 estimated variance을 구하는 방법을 제시한다. 그런데, 분산을 구하는 목적을 생각해보면 결국 statistical inference을 하기 위함인데, boostrap을 통해 구한 estimated variance을 이용하여 어떻게 신뢰구간이나 검정을 하는지에 대해서 공부를 해봐야 할 것 같다.