HW4_Variations on Logistic Regression

Eom SangJun

2020 10 9

#1. Latent Variables

어떤 학생의 능력을 T 라고 하고, 특정 문제의 난이도를 d 라고 하자. 이 때 T>d 일 때만 학생은 정답을 맞출 수 있다. 이제 d 를 고정하고 T 를 random variable 로 두고 density 를 f, 그리고 distribution function 을 F 라고 하자. 이 때 학생이 답을 틀릴 확률은

$p = P(T \leq d) = F(d)$

라고 할 수 있으며, 이 때 T 를 latent variable 이라고 한다. 즉, 직접적으로 관찰되지 않지만, 우리가 관심이 있는 변수 또는 결과값에 영향을 미치는 변수이다.

Distribution of T 를 logistic 이라고 한다면,

$$F(y) = \frac{\exp(y-\mu)/\sigma}{1+\exp(y-\mu)/\sigma}$$

이 때, y=T, $\mu = E(T)$를 의미한다.

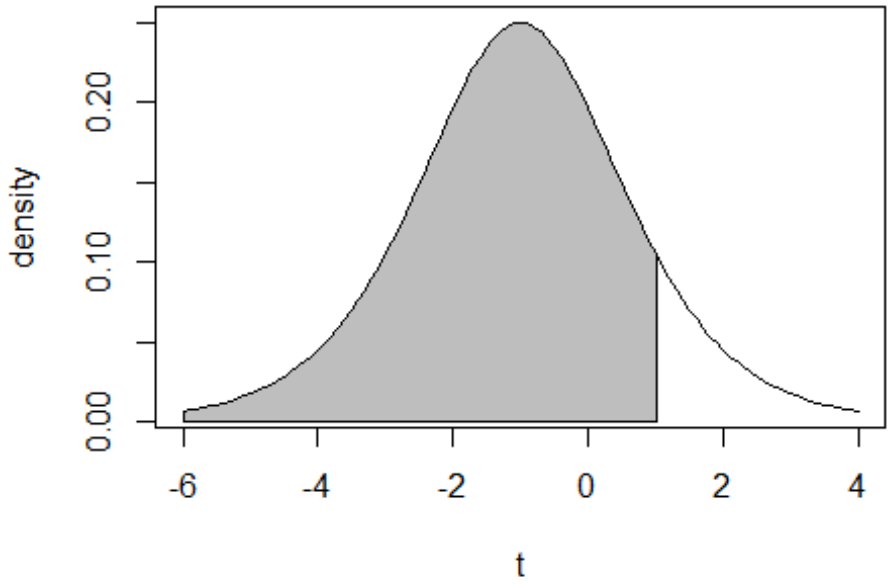
따라서

$$logit(p) = -\frac{\mu}{\sigma} + \frac{d}{\sigma}$$

만약 우리가 $\beta_0 = -\frac{\mu}{\sigma}, \beta_1 = \frac{d}{\sigma}$ 라고 둔다면 우리는 이제 logistic regression model 을 갖는 것이다.

d=1, $\sigma = 1$이고 T 가 -1 의 평균 값을 갖는다고 하자. 그 때 T 의 그래프를 그려보면 다음과 같다.

```
x<-seq(-6,4,0.1)
y<-dlogis(x,location = -1)
plot(x,y,type='l', ylab='density', xlab='t')
ii <- (x<=1)
polygon(c(x[ii], 1, -6), c(y[ii],0,0), col='gray')
```



➔ 그래프의 모양이 Normal Distribution 과 흡사하다.

➔ 회색으로 칠해진 부분이 F(d)의 부분으로 이 학생이 문제를 틀릴 확률을 나타낸다.

➔ 문제의 난이도는 1 인 반면에 T 의 평균값은 -1 이기 때문에 틀릴 확률이 절반 이상임을 알 수 있다.

#2. Link Functions

우리는 지금까지 logit link function 만을 probability 와 linear predictor 를 잇는 데에 사용하였다. 하지만 다른 선택지도 존재한다.

Latent Variable formulation 은 몇 개의 가능한 link function 들을 제시한다. 다음의 것들은 모두 glm function 내에 내장된 것들.

1. Probit: $\eta = \Phi^{-1}(p)$ 이 때 Φ는 normal cumulative distribution function. → Normally distributed latent variable 에서 비롯.

2. Complementary log-log: $\eta = \log(-\log(1-p))$ → Gumbel-distributed latent variable 에서 비롯.

3. Cauchit: $\eta = \tan^{-1}(\pi\left(p - \frac{1}{2}\right))$ → Cauchy-distributed latent variable 에서 비롯.

이제 각기 다른 insecticide concentrate level 에 따른 insects 죽음 여부에 대한 데이터를 가지고 각 link function 의 효과를 살펴보자.

```
data(bliss, package='faraway')
bliss
```

```
##   dead alive conc
## 1    2    28    0
## 2    8    22    1
## 3   15    15    2
## 4   23     7    3
## 5   27     3    4
```

→ Concentrate level 이 총 5 개이며 level 이 높아질수록 죽는 비율이 높아진다.

```
mlogit <- glm(cbind(dead, alive) ~ conc, family=binomial, data=bliss)
mprobit <- glm(cbind(dead, alive) ~ conc, family=binomial(link=probit), data=
bliss)
mcloglog <- glm(cbind(dead, alive) ~ conc, family=binomial(link=cloglog), dat
a=bliss)
mcauchit <- glm(cbind(dead, alive) ~ conc, family=binomial(link=cauchit), dat
a=bliss)
```

→ Link function 을 달리해서 model 을 만들어보자.

```
fitted(mlogit)
```

```
##         1         2         3         4         5
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823
```

```
predict(mlogit, type='response')
```

```
##         1         2         3         4         5
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823
```

```
coef(mlogit)[1] + coef(mlogit)[2]*bliss$conc
```

```
## [1] -2.323790e+00 -1.161895e+00  1.332268e-15  1.161895e+00  2.323790e+00
```

```
predict(mlogit)
```

```
##             1             2             3             4             5
## -2.323790e+00 -1.161895e+00  1.332268e-15  1.161895e+00  2.323790e+00
```

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 3.6.3
```

```
ilogit(mlogit$lin)
```

```
##           1           2           3           4           5
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823
```

→ logit link 를 사용했을 때의 각 level 에서의 예측 확률값은 위와 같다.

이제 같은 방식으로 logit link 부터 cauchit link 까지 각 level 에서의 예측 확률값들을 비교해보자.

```
predval <- sapply(list(mlogit, mprobit, mcloglog, mcauchit), fitted)
dimnames(predval) <- list(0:4, c('logit', 'probit', 'cloglog', 'cauchit'))
round(predval,3)
```

```
##   logit probit cloglog cauchit
## 0 0.089  0.084   0.127   0.119
## 1 0.238  0.245   0.250   0.213
## 2 0.500  0.498   0.455   0.506
## 3 0.762  0.752   0.722   0.791
## 4 0.911  0.914   0.933   0.882
```

→ 많이 차이가 나지 않는다는 것을 알 수 있다.

→ level 의 범위를 늘려서 차이가 뚜렷이 나타나도록 해보자.

```
dose <- seq(-4, 8, 0.2)
predval <- sapply(list(mlogit, mprobit, mcloglog, mcauchit), function(m)
  predict(m, data.frame(conc=dose), type='response'))
colnames(predval) <- c('logit', 'probit', 'cloglog', 'cauchit')
predval <- data.frame(dose, predval)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

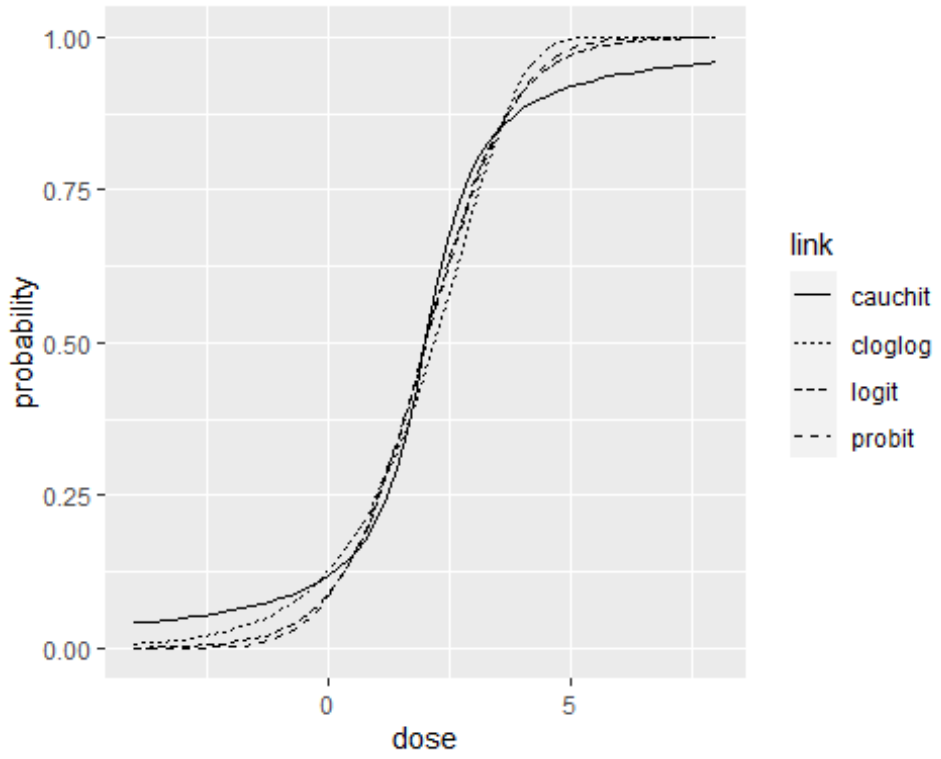
```
mpv <- gather(predval, link, probability, -dose)
```

→ tidyr package 의 gather function 은 기존의 data frame 을 key 값을 기준으로 새로운 형태로 정렬시켜준다.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
ggplot(mpv, aes(x=dose, y=probability, linetype=link))+geom_line()
```



➔ 0~5 사이에서는 차이가 크지 않지만 양 끝으로 갈수록 차이가 많이 난다는 것을 알 수 있다.

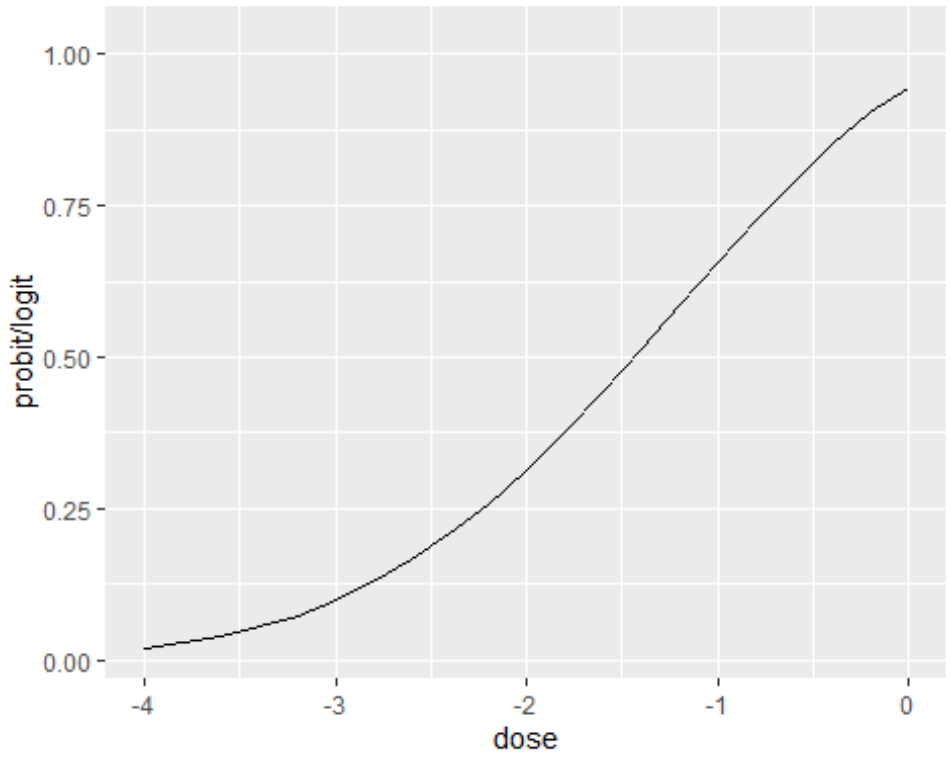
➔ 그 중 Cauchit 가 다른 세 개의 link function 들에 비해 가장 다른 형태를 보이는데 이는 latent variable 이 가장 가변적(variable)이기 때문이다. 양 끝에서 0 과 1 에 가장 천천히 converge 하는 모습을 보인다.

➔ Complementary log log 도 logit 과 probit 에 비해서는 차이를 보이는데 logit 과 probit 은 거의 동일해보인다.

➔ 우리는 lower 와 upper tail 에서의 ratio of probabilities 를 조사함으로써 logit 과 probit 의 차이를 알 수 있다.

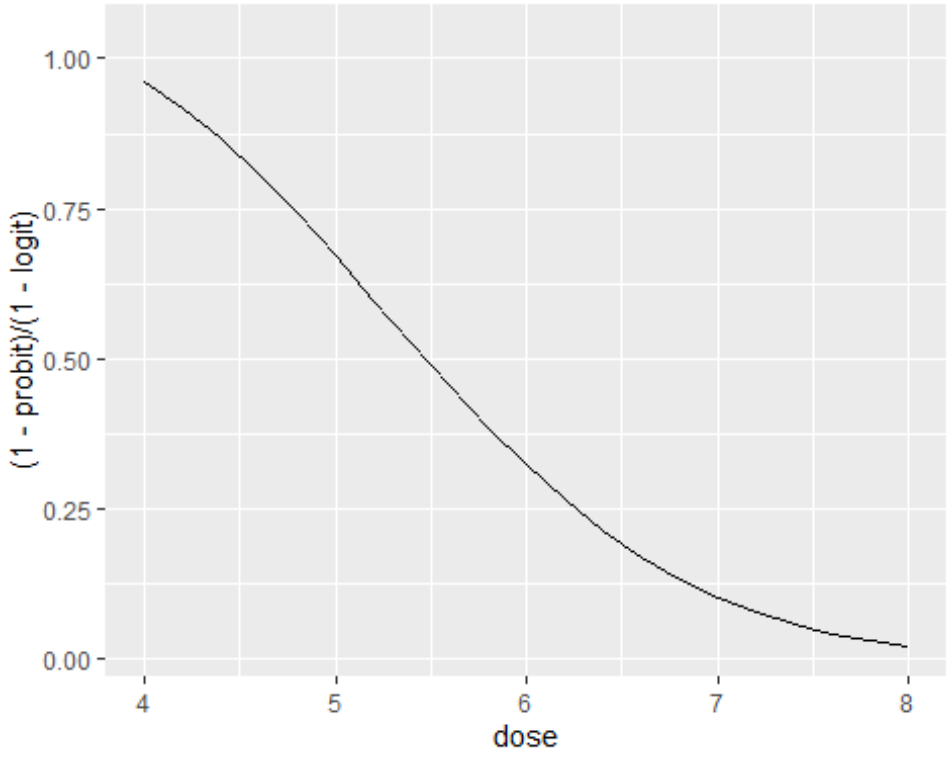
```
ggplot(predval, aes(x=dose, y=probit/logit)) + geom_line() + xlim(c(-4,0))
```

```
## Warning: Removed 40 row(s) containing missing values (geom_path).
```




```
ggplot(predval, aes(x=dose, y=(1-probit)/(1-logit))) + geom_line() + xlim(c(4,
8))

## Warning: Removed 40 row(s) containing missing values (geom_path).
```



➔ 만약 probit 과 logit 이 정말 차이가 없다면 비율은 1 에 가까워야 하지만 lower 과 upper tail 양 끝 쪽에 가까워질수록 1 에서 멀어지는 모습을 보인다. 즉, 양 끝 쪽에서 probit 과 logit 은 많은 차이를 보인다.

➔ 이는 Complementary log log 에서도 동일하게 나타난다.

그렇다면 어떤 link function 을 사용해야 하는가?

우선, 비용상의 문제 등 여러 한계로 인해 데이터만 가지고 link function 을 고르기란 쉽지 않다. 따라서 우리는 일반적으로 physical knowledge 등으로부터 가정된 사실들을 바탕으로 link function 을 고르거나 아니면 편의상 logit link 를 고른다.

Logit link 의 장점은

1. probit 보다 수학적으로 계산이 쉽다.

2. odds 를 사용하기 때문에 해석이 쉽다.

3. retrospectively sampled data 의 분석을 쉽게 해준다.

#3. Prospective and Retrospective Sampling

아기들에게 음식을 주는 방식과 respiratory disease 간의 관계에 대한 데이터를 통해 prospective sampling 과 retrospective sampling 간의 차이를 알아보자.

```
xtabs(disease/(disease+nondisease) ~ sex + food, babyfood)
```

```
##       food
## sex        Bottle     Breast      Suppl
##   Boy  0.16812227 0.09514170 0.12925170
##   Girl 0.12500000 0.06681034 0.12598425
```

성별과 음식을 주는 방식에 따른 질병 발생 비율은 위와 같다.

1.prospective sampling 에서는 predictor 들이 고정되어 있고 outcome 들이 관찰된다. 이를 cohort study 라고 한다. 위의 예시에서는 음식을 주는 방식을 고정시키고 질병 발생의 비율이 어떻게 되는 지를 관찰했다면 이는 prospective sampling 이다.

2. retrospective sampling 에서는 반대로 outcome 들이 고정되어 있고 predictor 들이 관찰된다. 이를 case-control study 라고 한다. 위의 예시에서는 병이 걸린 아기들과 그렇지 않은 아이들의 sample 을 구한 뒤 그들의 정보를 조사했다면 이것은 retrospective sampling 이다.

우리는 predictor 들이 response 에 어떻게 영향을 준 것인지가 궁금한 것이므로 prospective sampling 이 요구될 것으로 보인다. 우선 남아이고 bottle 과 breast feeding 경우만 살펴보자.

```
babyfood[c(1,3),]
```

```
##   disease nondisease sex   food
## 1      77        381 Boy Bottle
## 3      47        447 Boy Breast
```

log-odds 를 사용한다면, outcome 과 predictor 간의 연관성이 얼마나 강한지를 알 수 있을 것이다.

1. Breast Feeding 이 주어졌을 때, respiratory disease 를 가질 log-odds 는: log(47/447)=-2.25

2. Bottle Feeding 이 주어졌을 때, respiratory disease 를 가질 log-odds 는: log(77/381)=-1.60

→ 두 log-odds 의 차이 = log-odds ratio = -1.6-(-2.25)=0.65

즉, bottle fed 의 경우가 breast 에 비해 disease 에 대한 위험성이 더 크다.

그럼 retrospective sampling 관점에서는 어떠할까?

신기하게도, disease 가 주어졌을 때를 가정해도 log-odds 는 동일하다.

Δ=log77/47-log381/447=log77/381-log47/447=0.65

이는 retrospective design 이 prospective design 만큼 log-odds ratio 를 추정하는 데에 효과적이라는 것을 보여준다. 다만, probit 과 같은 다른 link 에 대해서는 불가능하고 오직 logit link 에서만 가능하다.

Retrospective design 의 장점

1. cohort 는 observation 들을 추적해야 해서 시간이 오래 걸리고, 비용이 많이 들지만, retrospective 는 그렇지 않다.

2. predictor 가 될 수 있는 것들을 많이 조사할 수 있다.

3. cohort 는 rare outcome 을 얻기 위해서는 매우 많은 양의 데이터를 필요로 할 수 있는 반면, retrospective 는 그렇지 않다.

Prospective design 의 장점

1. sample 을 고르는 데에 있어서 bias 가 개입할 가능성이 적다.

2. case-control 에서는 주로 historical records 를 참조하는데 이는 부정확하거나 불완전할 수 있다. Prospective 는 이런 문제가 발생할 가능성이 적다.

3. 하나보다 더 많은 outcome 의 study 를 가능하게 해준다.

4. outcome 의 확률을 계산할 수 있게 해준다.

어째서 prospective design 에서만 outcome 의 확률을 예측할 수 있는 지에 대해 알아보자.

π_0를 병을 가지고 있지 않은 사람이 study 에 포함될 확률, π_1를 병을 가진 사람이 study 에 포함될 확률이라고 하자. Prospective design 에서는 outcome 에 대한 지식이 없기 때문에 $\pi_0 = \pi_1$이라고 본다. 반면, retrospective 의 경우 일반적으로 π_0이 π_1보다 훨씬 낮다.

이번에는 $p^*(x)$를 어떤 사람이 study 에 포함되었을 때(given), 그 사람이 병을 가지고 있을 조건부 확률이라고 하자. 또한 $p(x)$를 어떤 사람이 병을 가지고 있을 marginal probability 또는 비조건부 확률이라 하자.

Bayes Theorem 에 의해

$$p^*(x) = \frac{\pi_1 p(x)}{\pi_1 p(x) + \pi_0(1 - p(x))}$$

따라서

$$\text{logit}\big(p^*(x)\big) = \log\left(\frac{\pi_1}{\pi_0}\right) + \text{logit}(p(x))$$

즉, retrospective 와 prospective 의 차이는 오직 $\log\left(\frac{\pi_1}{\pi_0}\right)$라는 것을 알 수 있다. 그런데 일반적으로 $\log\left(\frac{\pi_1}{\pi_0}\right)$는 알려져 있지 않다. 따라서 retrospective 에서는 covariates 의 relative effect 를 알 수는 있지만, 절대적인 effect 값을 알지는 못한다. 반면 prospective 는 $\pi_0 = \pi_1$ 이므로 절대적인 값을 계산할 수 있다.

#4. Prediction and Effective Doses

우리는 covariates 값이 주어졌을 때, outcome 을 예측하고 싶을 수 있다. 예를 들어, binomial case 에서는 성공확률을 예측하고 싶을 수 있다. 이 때 normal approximation 을 통해 confidence interval 을 구할 수도 있고, linear predictor 를 inverse of the link function 에 집어넣어서 확률 값을 점 추정할 수도 있다.

앞서 보았던 Insect data 를 이용하여 이를 살펴보자.

```
lmod <- glm(cbind(dead, alive) ~ conc, family=binomial, data=bliss)
lmodsum <- summary(lmod)
```

dose 가 2.5 일 때 insect 가 죽을 확률을 구해보자.

```
x0 <- c(1,2.5)
eta0 <- sum(x0*coef(lmod))
ilogit(eta0)
```

```
## [1] 0.6412854
```

→ 64% 확률로 죽을 것이다.

이번에는 Confidence Interval 을 구해보자. 우선, 이를 위해서는 variance matrix 를 구해야 한다.

```
(cm <- lmodsum$cov.unscaled)
```

```
##             (Intercept)        conc
## (Intercept)  0.17463024 -0.06582336
## conc        -0.06582336  0.03291168
```

따라서 logit scale(linear predictor)의 standard error 는 다음과 같다.

```
se <- sqrt(t(x0) %*% cm %*% x0) #%*%은 행렬곱, 내적임.
```

이에 따라 probability scale 상의 CI 를 구하면,

```
ilogit(c(eta0-1.96*se, eta0+1.96*se))
```

```
## [1] 0.5342962 0.7358471
```

이렇게 하는 것이 번거로울 경우 predict command 를 이용하면 쉽게 점 추정 값과 standard error 값을 구할 수 있다.

```
predict(lmod, newdata=data.frame(conc=2.5), se=T)
```

```
## $fit
##         1
## 0.5809475
##
## $se.fit
## [1] 0.2262995
##
## $residual.scale
## [1] 1
```

```
ilogit(c(0.58095-1.96*0.2263, 0.58095+1.96*0.2263))
```

```
## [1] 0.5342966 0.7358478
```

→ 위에서 계산한 것과 거의 동일한 결과 값을 얻을 수 있었다.

Linear Regression 상황과 달리 binomial 에서는 future observation 의 CI 와 mean response 의 CI 간 차이가 없다.

우리는 이제 dose 가 -5 일 때를 살펴보자.

```
x0 <- c(1,-5)
se <- sqrt(t(x0) %*% cm %*% x0)
eta0 <- sum(x0*lmod$coef)
ilogit(c(eta0-1.96*se, eta0+1.96*se))
```

```
## [1] 2.357639e-05 3.643038e-03
```

절대적인 CI interval 은 매우 작아보이지만, upper limit 이 lower limit 의 100 배가 넘는다. 즉 상대적으로 봤을 때 넓다고 할 수 있다.

이번에는 p 값이 고정되어 있고 그에 해당하는 covariates x 값을 찾고 싶다고 하자. 예를 들어서 p=1/2 일 때의 dose 값을 구하고 싶다고 하자. 이 때 그러한 dose 는 ED50 으로 표현한다(Effective Dose). 또는 어떤 대상을 죽이거나 하는 상황에서는 LD50 으로 표현하기도 한다(Lethal Dose).

P=1/2 일 때, logit(p) 값은 0 이 되고 이에 따라

$$\widehat{ED50} = -\hat{\beta}_0/\hat{\beta}_1$$

```
(ld50 <- -lmod$coef[1]/lmod$coef[2])
```

```
## (Intercept)
##           2
```

→ 우리의 데이터에서는 dose 가 2 일 때, 50% 확률 값을 가진다.

Standard Error 를 구하기 위해서 delta method 를 이용해보자.

Multivariate θ에 관하여 variance of g($\hat{\theta}$)의 일반적인 표현은 다음과 같다.

$var\ \mathrm{g}(\hat{\theta}) \approx \mathrm{g}'(\hat{\theta})^T\ var\ \hat{\theta}\ \mathrm{g}'(\hat{\theta})$

이를 이용했을 때 standard error 값은 다음과 같다.

```
dr <- c(-1/lmod$coef[2], lmod$coef[1]/lmod$coef[2]^2)
sqrt(dr %*% lmodsum$cov.unscaled %*% dr)[,]
```

```
## [1] 0.1784367
```

따라서 95% CI 값은,

```
c(2-1.96*0.178, 2+1.96*0.178)
```

```
## [1] 1.65112 2.34888
```

50%가 아니라 다른 level 에 대해서도 궁금할 수 있다.

이 때 effective dose 값은

$$x_p = \frac{logit(p) - \beta_0}{\beta_1}$$

90%일 때의 effective dose 값을 구해보자.

```
(ed90 <- (logit(0.9) - lmod$coef[1])/lmod$coef[2])
```

```
## (Intercept)
##     3.89107
```

MASS Package 에는 편리하게도 effective dose 값을 구해주는 dose.p function 이 있다.

```r
library(MASS)
dose.p(lmod, p=c(0.5, 0.9))
```

```
##             Dose        SE
## p = 0.5: 2.00000 0.1784367
## p = 0.9: 3.89107 0.3449965
```

#5. Matched Case-Control Studies

Case-Control Study 에서 우리는 outcome 에 대한 특정 risk factor 의 영향력(effect)를 알아내고자 한다. 그런데, 우리는 outcome 에 다른 confounding variable 들이 영향을 줄 수 있다는 것을 알고 있다. 따라서 이를 해결할 수 있는 한 가지 방법은 그 confounding variable 을 찾아서 모델에 집어넣는 것이다. 그러나 confounding variable 의 형태가 모델에 적합하지 않는 등의 문제가 있을 수 있다.

Matched case-control study 는 이러한 문제를 보완하기 위한 방법이다. 이는 각각의 case 를 하나 또는 더 많은 control 과 match 시키는 것인데 이 때 control 은 case 와 비교했을 때 potential confounding variable 의 관점에서 유사하거나 동일해야 한다. 그리고 이렇게 match 한 그룹을 우리는 matched-set 이라고 부른다. 이러한 matching 의 효과는 우리가 측정하기 어려운 confounder 들을 조정해주는 것이다.

당연히 confounding variable 을 더 많이 특정할수록 case 와 control 을 match 시키는 것은 더욱 어려워진다. 따라서 matching requirements 를 적절하게 조절할 필요가 있다.

다만 matched case-control study 에도 문제점들이 있는데, 우선 matched set 을 구성하는 것이 쉽지 않으며 match 를 하는 데에 사용한 variable 의 effect 를 측정할 수 없다는 단점이 있다. 또한 이렇게 match 를 시키면, random sampling 의 효과가 사라지기 때문에 relative effects 를 찾더라도 population group 으로 확장할 수 없게 된다.

때때로 case 는 rare 한데, control 은 쉽게 구할 수 있는 경우가 있다. 1:M design 은 각각의 case 에 대해 M 개의 control 이 있는 상황이다. 이 때 M 은 일반적으로 작지만, matched set 에 따라 그 크기가 매우 다양할 수 있다. 각 추가적인 control 은 risk factor 를 estimate 하는 데에 있어서 increased efficiency 를 오히려 감소시키는 결과를 가져오므로 M=5 를 초과하는 것은 좋지 않다.

이제 logistic regression model 을 세워보자.

i 를 individual, j-th matched set 에 대해 covariate vector x_{ij}를 생각해보자. x_{ij}는 우리가 관심있는 risk factor 뿐만 아니라 우리가 adjust 하고 싶지만, 모종의 이유로 matched set 을 만들 때 criteria 로 만들지는 못한 variable 까지 포함할 것이다. Matched set 은 총 n 개가 있고 i=0 을 case 로, i=1,...,M 을 control 이라고 하자. 그럼 이 때 logistic regression model 의 form 은 다음과 같다.

$$logit\left(p_j(x_{ij})\right) = \alpha_j + \beta^T x_{ij}$$

α_j는 j 번째 matched set 안에서 confounding variables 의 effect 를 의미한다. 그리고 이 때 conditional probability of the observed outcome 은 다음과 같다.

$$\frac{\exp \beta^T x_{0j}}{\sum_{i=0}^{M} \exp \beta^T x_{ij}}$$

→ α_j가 사라진다는 것을 알 수 있다.

Conditional likelihood for the model 은 다음과 같다.

$$L(\beta) = \prod_{j=1}^{n} \left\{ 1 + \sum_{i=1}^{M} \exp[\beta^T (x_{ij} - x_{0j})] \right\}^{-1}$$

우리는 이제 inference 를 위해 standard likelihood methods 를 사용할 것이다. 위의 Likelihood form 은 생존분석에서 사용되는 proportional hazards model 에 대한 likelihood 와 형태가 동일하다. 따라서 우리는 이를 이용할 것이다.

참고로 α_s에 대한 추정이 안 되기 때문에, 개개인의 prediction 값은 구할 수 없다. 단지, βs 에 의해서 측정되는 relative risk 만 구할 수 있다.

X-ray 와 childhood acute myeloid leukemia 간의 관계에 대한 데이터로 앞서 언급했던 내용들을 살펴보자.

```
head(amlxray)
```

```
##     ID disease Sex downs age Mray MupRay MlowRay Fray Cray CnRay
## 1 7004       1   F    no   0   no     no      no   no   no     1
## 2 7004       0   F    no   0   no     no      no   no   no     1
## 3 7006       1   M    no   6   no     no      no   no  yes     3
## 4 7006       0   M    no   6   no     no      no   no  yes     2
## 5 7009       1   F    no   8   no     no      no   no   no     1
## 6 7009       0   F    no   8   no     no      no   no   no     1
```

→ Case 한 개와 Control 한 개로 짝 지어진 matched set 을 볼 수 있다. 앞에 나온 것이 case, 뒤에 나온 것이 control

→ 여기서 나이는 단지 matching variable 로서 사용되었다. 그리고 나머지 변수들은 모두 risk factor 로 사용되었다.

→ 그런데 Downs syndrome 은 risk factor 로 이미 알려져 있다. 그리고 subjects 중 오직 7 개만 down syndrome 에 해당한다는 것을 알 수 있다.

```
amlxray[amlxray$downs=='yes', 1:4]
```

```
##       ID disease Sex downs
## 7   7010       1   M   yes
## 17  7018       1   F   yes
## 78  7066       1   F   yes
## 88  7077       1   M   yes
## 173 7146       1   F   yes
```

```
## 196 7176       1   F   yes
## 210 7189       1   F   yes
```

→ Down syndrome 을 가지고 있는 경우 모두 case 에 해당했다. 따라서 만약 이를 variable 로 넣는다면, coefficient 값이 무한대로 발산할 것이다. 따라서 이를 제외해주자.

```
(ii <- which(amlxray$downs=='yes'))
```

```
## [1]   7  17  78  88 173 196 210
```

```
ramlxray <- amlxray[-c(ii,ii+1),]
```

추가적으로 Mray, MupRay, MlowRay 의 경우 각각 아이의 어머니가 한 번이라도 X-ray 를 촬영했거나 upper body X-ray 를 촬영했거나, lower body x-ray 를 촬영했거나를 나타낸다. 이 변수들은 모두 상당히 관련이 있기 때문에 일단은 그냥 Mray 만을 선택하기로 한다.

CnRay 와 Cray 의 경우도 CnRay 는 아이가 X-ray 를 정확히 몇 번을 찍었는 지를 알려주는 반면, Cray 는 단지 찍은 경험이 있는 지만을 알려주므로 CnRay 만을 선택한다.

Survival Package 에는 proportional hazards model 을 만들 수 있는 함수가 존재한다.

```
library(survival)
```

```
##
## Attaching package: 'survival'
```

```
## The following objects are masked from 'package:faraway':
##
##     rats, solder
```

```
cmod <- clogit(disease ~ Sex+Mray+Fray+CnRay+strata(ID),ramlxray)
```

→ conditional logit model 을 fit 하기 위해서는 clogit function 을 사용해야 한다.

→ matched set 의 경우 반드시 독립변수 쪽에 strata function 을 사용해서 표시를 해주어야 한다.

```
summary(cmod)
```

```
## Call:
## coxph(formula = Surv(rep(1, 224L), disease) ~ Sex + Mray + Fray +
##     CnRay + strata(ID), data = ramlxray, method = "exact")
##
##   n= 224, number of events= 104
##
```

```
##             coef exp(coef) se(coef)      z Pr(>|z|)
## SexM     0.1563    1.1691   0.3861  0.405  0.68566
## Mrayyes  0.2276    1.2556   0.5821  0.391  0.69573
## Frayyes  0.6933    2.0003   0.3512  1.974  0.04839 *
## CnRay.L  1.9408    6.9641   0.6207  3.127  0.00177 **
## CnRay.Q -0.2480    0.7803   0.5819 -0.426  0.66993
## CnRay.C -0.5801    0.5599   0.5906 -0.982  0.32598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## SexM        1.1691     0.8553    0.5486     2.492
## Mrayyes     1.2556     0.7964    0.4013     3.929
## Frayyes     2.0003     0.4999    1.0049     3.982
## CnRay.L     6.9641     0.1436    2.0631    23.507
## CnRay.Q     0.7803     1.2815    0.2495     2.441
## CnRay.C     0.5599     1.7862    0.1759     1.781
##
## Concordance= 0.662  (se = 0.056 )
## Likelihood ratio test= 20.89  on 6 df,   p=0.002
## Wald test            = 14.49  on 6 df,   p=0.02
## Score (logrank) test = 18.6  on 6 df,   p=0.005
```

→ 성별과 Mray 는 통계적으로 유의하지 않은 것으로 보인다.

→ Overall Test 의 경우 적어도 어떤 한 variable 은 통계적으로 유의함을 나타내주는 것이다. P-value 들이 전부 0.05 보다 낮기 때문에 적어도 하나의 variable 은 통계적으로 유의하다.

→ Fray 와 CnRay 는 통계적으로 유의한데, 그 중 CnRay.L 이 가장 명확하게 통계적으로 유의하다고 드러난다.

→ CnRay 는 ordered Factor 로, linear, quadratic, cubic contrast 를 사용하는데, 이 중 오직 linear effect 만 significant 하다.

→ CnRay 의 linear effect 만 significant 하기 때문에 ordered factor 인 CnRay 를 numeric data 로 변환해서 분석을 다시 해보자. 그리고 유의하지 않았던 변수들도 빼서 진행해보자.

```
cmodr <- clogit(disease ~ Fray + unclass(CnRay)+strata(ID), ramlxray)
summary(cmodr)
```

```
## Call:
## coxph(formula = Surv(rep(1, 224L), disease) ~ Fray + unclass(CnRay) +
##     strata(ID), data = ramlxray, method = "exact")
##
##   n= 224, number of events= 104
##
##                  coef exp(coef) se(coef)     z Pr(>|z|)
```

```
## Frayyes         0.6704    1.9550   0.3441 1.948 0.051394 .
## unclass(CnRay) 0.8145    2.2580   0.2368 3.439 0.000584 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## Frayyes            1.955     0.5115     0.996     3.838
## unclass(CnRay)     2.258     0.4429     1.419     3.592
##
## Concordance= 0.654  (se = 0.052 )
## Likelihood ratio test= 19.55  on 2 df,   p=6e-05
## Wald test            = 14.12  on 2 df,   p=9e-04
## Score (logrank) test = 17.56  on 2 df,   p=2e-04
```

→ CnRay 의 값은 그대로 숫자로 대응되는 것이 아니라, 1=none, 2=1 or 2 x-rays, 3=3 or 4 x-rays, 4=5 or more x-rays 라는 것을 유의하자.

→ 그럼 이 때 인접한 category 로 이동할 때(이 때 이동은 1 에서 2, 2 에서 3 등 올라가는 것) odds of the disease 는 2.26 상승한다.

→ 유의할 것은 Fray 가 이번에는 통계적으로 유의하지 않다고 나왔다는 점이다.

```
gmod <- glm(disease ~ Fray + unclass(CnRay), family=binomial, ramlxray)
sumary(gmod)
```

```
##                Estimate Std. Error z value  Pr(>|z|)
## (Intercept)    -1.16228    0.30105 -3.8607 0.0001131
## Frayyes         0.50035    0.30780  1.6255 0.1040461
## unclass(CnRay)  0.60054    0.17739  3.3855 0.0007106
##
## n = 224 p = 3
## Deviance = 293.26338 Null Deviance = 309.38611 (Difference = 16.12272)
```

→ 위의 분석은 흔히 실수할 수 있는 분석방식이다. 다른 결과를 보여준다.

우리가 비록 child 가 x-ray 를 찍는 것에 대한 영향을 찾아냈지만, 우리는 x-ray 가 disease 의 원인이라고 단정할 수 없다. 왜냐하면 무언가 문제가 있는 사람들만 보통 x-ray 를 찍기 때문에, x-ray 는 모종의 다른 원인적 변수와 관련이 있을 가능성이 높다.