Test-Implementation-05-31-2021

Sixtus Dakurah

5/31/2021

Define the variables:

B : Total amount available for disbursement.

$S_i, \quad i = 1, 2, ..., s$: Denote segment i with a total of s segments.

L : Loan amount.

P : The price variable.

Step 1: Fit the logistic model for a given segment. Here the segment indexes will be omitted and all variables will be assumed to pertain to a single segment.

The design matrix is of the form:

$X = [j, X_1, X_2, X_3]$ where $X_l = [x_{1,l}, ..., X_{m,l}]'$ $l = 1$: Segment, $l = 2$: Price, $l = 3$: Loan Amount.

The model for predicting the probability of booking is of the form:

$$\pi(X, \beta) = [\sigma(z_i), ..., \sigma(z_m)]' \tag{1}$$

Where $\sigma(z_k) = [1 + exp\{-z_k\}]^{-1}$ and $z_k = X_k\beta$, $k = 1, ..., m$ the number of examples.

An expressive form for $X_k\beta = \beta_0 + \beta_1 x_{k,1} + \beta_2 x_{k,2} + \beta_3 x_{k,3} + \beta_{1,2} x_{k,1} * x_{k,2} + \beta_{1,3} x_{k,1} * x_{k,3}$

We now simulate some data:

```
set.seed(111)
# set the number of examples and segments
m <- 300; m1 <- 100; m2<-100; m3 <- 100; s <- 3
# create the segment groups
x1 <- c(rep('s1', m1), rep('s2', m2), rep('s3', m3))
# generate the prices
x2 <- c(
  rlnorm(m1, meanlog = 1, sdlog = 1),
  rlnorm(m2, meanlog = 3, sdlog = 1),
  rlnorm(m3, meanlog = 5, sdlog = 1)
  )
# generate the loan amount
x3 <- c(
  rep(rlnorm(1, meanlog = 11, sdlog = 5), m1),
  rep(rlnorm(1, meanlog = 3, sdlog = 3), m2),
  rep(rlnorm(1, meanlog = 15, sdlog = 7), m3)
  )
# generate the prob
prob.level1 <- function(val){
```

```r
  ifelse(val < quantile(x2[1:m1], 0.25), runif(1, 0.4, 1), runif(1, 0, 1)) }
prob.level2 <- function(val){
  ifelse(val < quantile(x2[(m1+1):(m1+m2)], 0.25), runif(1, 0.4, 01), runif(1, 0, 1)) }
prob.level3 <- function(val){
  ifelse(val < quantile(x2[(m1+m2+1):m], 0.25), runif(1, 0.4, 01), runif(1, 0, 1)) }
probs <- c(

  do.call(rbind, lapply(x2[1:m1], prob.level1)),
  do.call(rbind, lapply(x2[(m1+1):(m1+m2)], prob.level2)),
  do.call(rbind, lapply(x2[(m1+m2+1):m], prob.level3))

)
data.sim <- data.frame(x1, x2, x3, probs,
                       y = as.factor(ifelse(probs > 0.5, 's', 'f')))
# the price can not be more than the loan amount
data.sim <- data.sim %>% mutate(price = ifelse(x2 > x3, 10*x3, x3))

head(data.sim)
```

```
##   x1        x2     x3      probs y  price
## 1 s1 3.4391375 122474 0.03215337 f 122474
## 2 s1 1.9527998 122474 0.83040273 s 122474
## 3 s1 1.9904807 122474 0.43669723 f 122474
## 4 s1 0.2718933 122474 0.55923602 s 122474
## 5 s1 2.2913106 122474 0.71349150 s 122474
## 6 s1 3.1276384 122474 0.91685147 s 122474
```

```r
summary(data.sim)
```

```
##       x1                  x2                 x3              probs
##  Length:300         Min.   :   0.1208   Min.   :    59.8   Min.   :0.000673
##  Class :character   1st Qu.:   4.1426   1st Qu.:    59.8   1st Qu.:0.346875
##  Mode  :character   Median :  21.6488   Median :122474.0   Median :0.555124
##                     Mean   : 102.3133   Mean   :311193.2   Mean   :0.538676
##                     3rd Qu.:  96.6912   3rd Qu.:811045.9   3rd Qu.:0.775436
##                     Max.   :2040.9550   Max.   :811045.9   Max.   :0.998970
##  y           price
##  f:131   Min.   :    59.8
##  s:169   1st Qu.:    59.8
##          Median :122474.0
##          Mean   :311209.4
##          3rd Qu.:811045.9
##          Max.   :811045.9
```

We now build a logistic model using the data from segment 1

```r
data.segment1 <- data.sim %>% filter(x1 == 's1')
#glm.segment1.fit <- glm(y~x1 + x2 + x3 + I(x1*x2) +
#I(x1*x3), data = data.segment1, family = binomial)
glm.segment1.fit <- glm(y~x2, data = data.segment1, family = binomial)
summary(glm.segment1.fit)
```

```
##
## Call:
## glm(formula = y ~ x2, family = binomial, data = data.segment1)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.335  -1.292   1.025   1.057   1.455
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.38749    0.25623   1.512    0.130
## x2          -0.03121    0.03391  -0.920    0.357
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.19  on 99  degrees of freedom
## Residual deviance: 136.29  on 98  degrees of freedom
## AIC: 140.29
##
## Number of Fisher Scoring iterations: 4
```

The fitted model has the form:

$\hat{\sigma}(z_k) = [1 + exp\{0.38749 - 0.03121x_{k,2}\}]^{-1}$

$\pi(x_2, \hat{\beta}) = [\hat{\sigma}(z_1), ..., \hat{\sigma}(z_k)]'$

```
# get the predictions
segment1.pred.probs <- predict(glm.segment1.fit, type = "response")
data.segment1 <- data.segment1 %>%
  mutate(pred.probs = segment1.pred.probs,
         pred.y = as.factor(ifelse(pred.probs > 0.5, 's', 'f')))
head(data.segment1)
```

```
##   x1        x2     x3      probs y  price pred.probs pred.y
## 1 s1 3.4391375 122474 0.03215337 f 122474  0.5695847      s
## 2 s1 1.9527998 122474 0.83040273 s 122474  0.5809184      s
## 3 s1 1.9904807 122474 0.43669723 f 122474  0.5806321      s
## 4 s1 0.2718933 122474 0.55923602 s 122474  0.5936331      s
## 5 s1 2.2913106 122474 0.71349150 s 122474  0.5783442      s
## 6 s1 3.1276384 122474 0.91685147 s 122474  0.5719664      s
```

```
table(data.segment1$pred.y, data.segment1$y)
```

```
##
##      f  s
##   f  4  3
##   s 40 53
```

Step 2: We now optimize over the price variable.

```r
# Compute the total amount available for disbursement
(B <- sum(data.sim$x3))
```

```
## [1] 93357968
```

Denote the profit as:

$\rho(x_2, x_4) = x_2 - x_4$ where the new variable x_4 is the cost of booking.

The expected profit is then given as:

$$E\left[\rho(x_2, x_4)|x_{\{1,i\}}, x_2\right] = \sum_{k=1}^{m_i} \pi(x_{\{k,2\}}, \hat{\beta})\rho(x_{\{k,2\}}, x_{\{k,4\}}) \tag{2}$$

We also have the expected loan amount of the form:

$$E\left[x_3|x_{\{1,i\}}, x_2\right] = \sum_{k=1}^{m_i} \pi(x_{\{k,2\}}, \hat{\beta}) * x_{k,3} \tag{3}$$

Where $\pi(x_{\{k,2\}}, \hat{\beta})$ is assumed to be constructed for segment i.

The optimization problem is now of the form:

$$argmin_{x_2} E\left[\rho(x_2, x_4)|x_{\{1,i\}}, x_2\right] \qquad s.t. \quad \sum_{i=1}^{s} E\left[x_3|x_{\{1,i\}}, x_2\right] < B \tag{4}$$

The lagrangian is of the form:

$$F(x_{k,2}, \lambda) = E\left[\rho(x_2, x_4)|x_{\{1,i\}}, x_2\right] - \lambda\left[\sum_{i=1}^{s} E\left[x_3|x_{\{1,i\}}, x_2\right] - B\right] \tag{5}$$

Here the variable $x_{k,2}$ and x_2 will be used interchangeably, where the latter is preferred used and will be assumed to not depend (???) on k in both (2) and (3), otherwise it will be impossible to get Newton's update rules.

For simplicity let:

$f(x_{k,2}) = E\left[\rho(x_2, x_4)|x_{\{1,i\}}, x_2\right]$ and $c(x_{k,2}) = \sum_{i=1}^{s} E\left[x_3|x_{\{1,i\}}, x_2\right] - B$

Using newtons method, we have the following derivations:

Opt 1: Quadratic Taylor series expansion

$$F(x_{k,2}, \lambda) \approx F(x^o_{k,2}, \lambda^o) + (x_{k,2} - x^o_{k,2}) \left.\frac{\partial F}{\partial x_{k,2}}\right|_0 + (\lambda - \lambda^o) \left.\frac{\partial c}{\partial \lambda}\right|_0 + \frac{1}{2}(x_{k,2} - x^o_{k,2})^2 \left.\frac{\partial^2 F}{\partial x^2_{k,2}}\right|_0 + (x_{k,2} - x^o_{k,2}) * (\lambda - \lambda^o) \left.\frac{\partial^2 F}{\partial x_{k,2}\partial\lambda}\right|_0 \tag{6}$$

Opt 2: Inserting (5) into (6), we have

$$\begin{gathered} F(x_{k,2}, \lambda) \approx F(x^o_{k,2}, \lambda^o) + (x_{k,2} - x^o_{k,2}) \left\{ \left.\frac{\partial f}{\partial x_{k,2}}\right|_0 - \lambda^o \left.\frac{\partial c}{\partial x_{k,2}}\right|_0 \right\} - (\lambda - \lambda^o) c(x^o_{k,2}) + \\ \frac{1}{2}(x_{k,2} - x^o_{k,2})^2 \left\{ \left.\frac{\partial^2 f}{\partial x^2_{k,2}}\right|_0 - \lambda \left.\frac{\partial^2 c}{\partial x^2_{2,k}}\right|_0 \right\} - (x_{k,2} - x^o_{k,2}) * (\lambda - \lambda^o) \left.\frac{\partial c}{\partial x_{k,2}\partial\lambda}\right|_0 \end{gathered} \tag{7}$$

Opt 3: The maximum at $x_{k,2}$ is achieved when

$$\frac{\partial F}{\partial x_{k,2}} = \left.\frac{\partial f}{\partial x_{k,2}}\right|_0 + (x_{k,2} - x^o_{k,2})\left\{\left.\frac{\partial^2 f}{\partial x^2_{k,2}}\right|_0\right\} - \lambda \left.\frac{\partial c}{\partial x_{k,2}\partial\lambda}\right|_0 = 0 \tag{8}$$

Opt 3: We now derive the gradient update rules

We can easily see that:

$$\Delta x_{k,2} = \frac{\left\{\lambda \left.\frac{\partial c}{\partial x_{k,2}\partial\lambda}\right|_0 - \left.\frac{\partial f}{\partial x_{k,2}}\right|_0\right\}}{\left\{\left.\frac{\partial^2 f}{\partial x^2_{k,2}}\right|_0\right\}} \tag{9}$$

Similarly for a first order expansion of the constraints, we have the following new value of λ

$$c(x_{k,2}) = c(x^o_{k,2}) + \Delta x_{k,2} \left.\frac{\partial c}{\partial x_{k,2}}\right|_{x^o_{k,2}} = 0 \tag{10}$$

If we put (9) into (10), we can solve for λ as follows:

$$\lambda = \frac{\left\{\lambda \left.\frac{\partial c}{\partial x_{k,2}\partial\lambda}\right|_0\right\} * \left\{\left.\frac{\partial f}{\partial x_{k,2}}\right|_0\right\}}{\left\{\left.\frac{\partial c}{\partial x_{k,2}}\right|_0\right\}^2} - \frac{\left\{\left.\frac{\partial^2 f}{\partial x^2_{k,2}}\right|_0\right\} * c(x^o_{k,2})}{\left\{\left.\frac{\partial c}{\partial x_{k,2}}\right|_0\right\}^2} \tag{11}$$

Putting this into (10) will give final update rule.