Topic 5:

Segmentation through latent consumer types

Takeaki Sunada[1]

[1]Simon Business School
University of Rochester

A note

- A recent update of "mlogit" package has caused some compatibility issues with certain functionalities of gmnl functions.
- When you implement latent segment models (i.e. contents from this slide deck), please use the previous version of mlogit, whose "install.package" is available at the top of the lecture code.
- Please run Rstudio *as administrator* and run the install.package line. Once the installation is done, you should have version 1.0.2 installed (to check that, please run 'packageVersion("mlogit")').

Heterogeneous demand

- Last time we studied how we can segment consumers through their observed characteristics. That method requires that we have access to data on some consumer characteristics (e.g. demographics).
- Suppose now that we don't have access to it (how come we as a firm can get access to info about consumer's income in reality?) or demographic info does not provide any reasonable segmentation.
- Today we introduce another approach - "latent segment", to do segmentation even in the absence of demographic info.

Latent segment approach

- To the extent that we have access to panel data of consumer choices, we see history of each consumer's purchase decisions.
- "What they bought in the past" should be a good indicator of "what they like". Even in the absence of demographic information, we should be able to group together a set of individuals who exhibit similar purchase patterns in the past.
- Now that clustering is not through observed characteristics, but through consumer's latent preferences. Those who seem to have similar preference are to be grouped together - which is exactly what we want.
- Hence this approach is called "models with latent types (segments)".

Models with latent segments

- We assume that there are $\{1, ..., k, ..., K\}$ different types of consumers in the market, each with choice probability:

$$Pr_k(y = j \mid P) = \frac{\exp(\beta_{0k}^{j} + \beta_{1k} P^{j})}{1 + \sum_{j'=1}^{J} \exp(\beta_{0k}^{j'} + \beta_{1k} P^{j'})},$$

- Denote by w_k the proportion of type k consumers in the market. Because we cannot estimate w_k from demographic data anymore, w_k needs to be estimated by MLE along with β_{0k} and β_{1k}.
- All consumers are observationally identical, so we cannot tell which consumer is which type. Hence w_k is equivalent to the probability (to us) that each consumer is type k.

Latent segment approach

- We first implement models with latent segments in R using gmnl.
- We then discuss how gmnl estimates the parameters of latent segments - it's an application of MLE, but in a bit more complicated environment.
- Finally, we cover how to use the recovered demand system for pricing, and we compare latent demand models with demographic-based segmentation.

Implementation in R

```
#Run two-type model
lc2=gmnl(choice~price|1|0|0|1,data=mlogitdata,model='lc',Q=2,panel=TRUE)
summary(lc2)
```

- Let's estimate a model with two latent types. As usual, we use gmnl function.
- The syntax is "choice $\sim$ price $|1|0|0|1$". Changing this part would let us run more complicated models - we may not cover them this time (but note that the first argument was used in regression-in-logit models).
- We also add "model=lc" to indicate that we are estimating a model with latent types.
- "Q" assigns the number of latent segments.
- "panel" is always set to "true".

Implementation in R

```
Coefficients:
                          Estimate Std. Error z-value  Pr(>|z|)    
class.1.KB:(intercept)     5.17368    1.12831  4.5853 4.533e-06 ***
class.1.KR:(intercept)     2.77037    1.15097  2.4070 0.0160847 *  
class.1.MB:(intercept)     3.81657    1.13161  3.3727 0.0007444 ***
class.1.price             -4.40595    0.82333 -5.3514 8.728e-08 ***
class.2.KB:(intercept)     7.90490    1.26937  6.2274 4.742e-10 ***
class.2.KR:(intercept)     9.35652    1.33298  7.0193 2.230e-12 ***
class.2.MB:(intercept)     8.63305    1.21039  7.1325 9.859e-13 ***
class.2.price             -6.29007    0.92070 -6.8318 8.384e-12 ***
(class)2                  -0.11499    0.12320 -0.9333 0.3506591    
```

- gmnl estimates β^j_{0k} and β_{1k} for both types, as well as w_2: the proportion of type 2 consumers.
- β^j_{0k} and β_{1k} for each type can be interpreted in the same way as in the K-mean case. w_2 requires some extra calculation.

Implementation in R

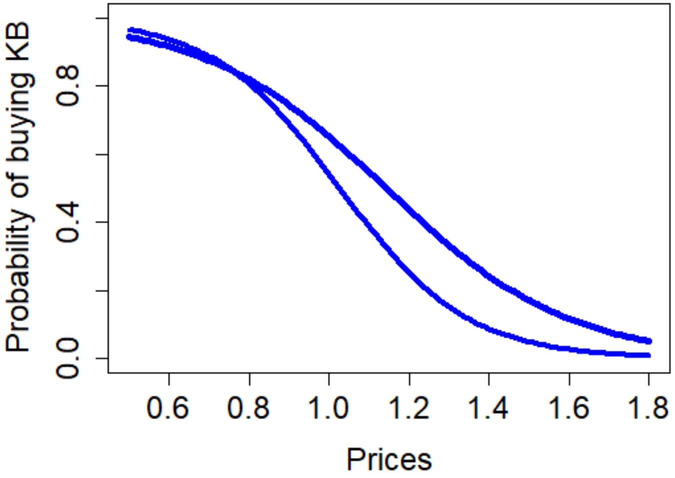
```
Coefficients:
                    Estimate Std. Error z-value  Pr(>|z|)
(class)2            -0.11499    0.12320 -0.9333 0.3506591
```

- gmnl does not report w_k in the form of proportion as is. Instead, we need to recover the proportion of each type by the following transformation. Denoting "(class)k" object from gmnl by γ_k:

$$w_k = \frac{\exp(\gamma_k)}{1 + \sum_{k'=2}^{K} \exp(\gamma_{k'})}, \text{for } k = 2, ..., K$$

  and $w_1 = 1 - \sum_{k=2}^{K} w_k$.
- In this example, $w_2 = \frac{\exp(-0.115)}{1+\exp(-0.115)} = 0.471$ and $w_1 = 1 - w_2 = 0.529$.

Demand with two-segment model



Implementation in R

```
#Run three type model
lc3=gmnl(choice~price|1|0|0|1,data=mlogitdata,model='lc',Q=3,panel=TRUE)
summary(lc3)
```

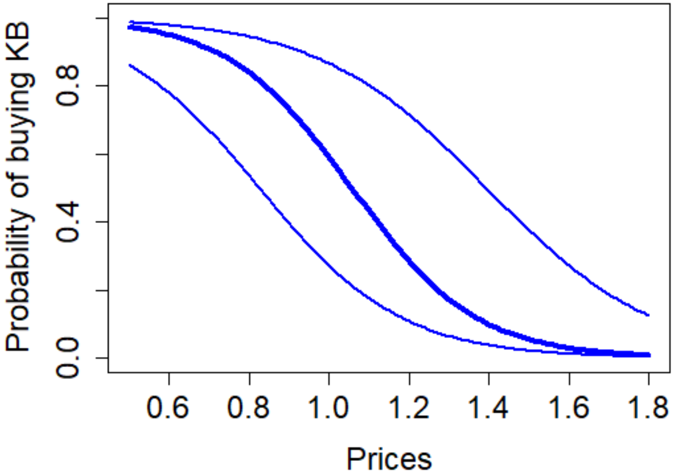
- In the same way, we can run a model with three types.

Implementation in R

```
Coefficients:
                          Estimate Std. Error z-value  Pr(>|z|)    
class.1.KB:(intercept)     4.75285    2.19368  2.1666 0.0302647 *  
class.1.KR:(intercept)     3.49627    2.14109  1.6329 0.1024807    
class.1.MB:(intercept)     5.44566    2.22897  2.4431 0.0145607 *  
class.1.price             -5.61169    1.67570 -3.3489 0.0008115 ***
class.2.KB:(intercept)     6.72616    2.33538  2.8801 0.0039753 ** 
class.2.KR:(intercept)     3.57800    2.09928  1.7044 0.0883076 .  
class.2.MB:(intercept)     3.58913    2.19093  1.6382 0.1013855    
class.2.price             -4.75358    1.63590 -2.9058 0.0036633 ** 
class.3.KB:(intercept)     8.16833    1.22179  6.6855 2.301e-11 ***
class.3.KR:(intercept)     9.42620    1.27381  7.4000 1.361e-13 ***
class.3.MB:(intercept)     8.77645    1.16804  7.5138 5.751e-14 ***
class.3.price             -6.41396    0.90081 -7.1202 1.078e-12 ***
(class)2                  -0.10470    0.16216 -0.6457 0.5184942    
(class)3                   0.61189    0.13668  4.4767 7.582e-06 ***
```

- This time, $w_2 = \frac{\exp(-0.105)}{1+\exp(-0.105)+\exp(0.612)} = 0.241$, $w_3 = \frac{\exp(0.612)}{1+\exp(-0.105)+\exp(0.612)} = 0.492$ and $w_1 = 1 - w_2 - w_3 = 0.267$.

Demand with three-segment model



Choosing the number of segments

- We can estimate models with any number of segments - how do we pick the right number of segments?
- This model is estimated by a *single* MLE. i.e. parameters of all segments are estimated by running gmnl once. Hence we can use BIC.

Choosing the number of segments

- BIC with no segmentation:

```
[1] 1060.024
```

- Two segments:

```
[1] 965.4554
```

- Three:

```
[1] 961.7133
```

- Four:

```
[1] 942.2066
```

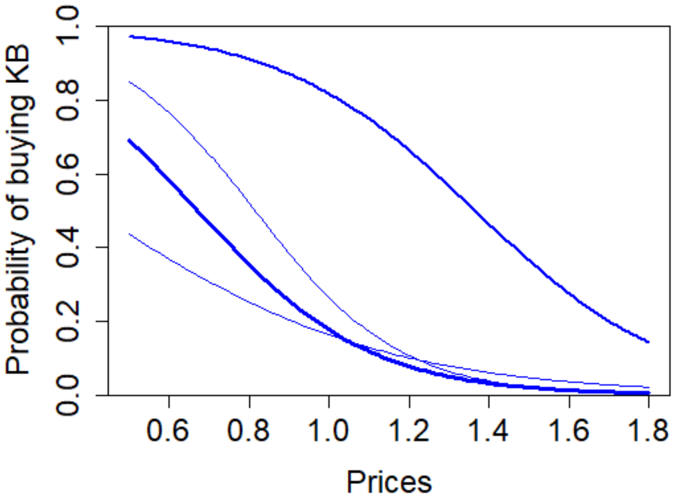
- Five:

```
[1] 981.373
```

Estimated parameters of four-segment model

```
Coefficients:
                        Estimate Std. Error z-value  Pr(>|z|)    
class.1.KB:(intercept)   4.57734    1.68285  2.7200 0.0065284 ** 
class.1.KR:(intercept)   2.30125    1.71091  1.3450 0.1786098    
class.1.MB:(intercept)   4.88573    2.17868  2.2425 0.0249277 *  
class.1.price           -5.52225    1.67580 -3.2953 0.0009832 ***
class.2.KB:(intercept)   3.86201    1.31516  2.9365 0.0033190 ** 
class.2.KR:(intercept)   4.81445    0.89045  5.4068 6.417e-08 ***
class.2.MB:(intercept)   6.13200    0.91236  6.7210 1.805e-11 ***
class.2.price           -4.69838    0.68504 -6.8585 6.958e-12 ***
class.3.KB:(intercept)   6.22360    1.18477  5.2530 1.496e-07 ***
class.3.KR:(intercept)   4.28802    1.14026  3.7605 0.0001695 ***
class.3.MB:(intercept)   4.99226    1.24903  3.9969 6.418e-05 ***
class.3.price           -4.11882    0.98987 -4.1610 3.169e-05 ***
class.4.KB:(intercept)   3.88780    1.56668  2.4815 0.0130813 *  
class.4.KR:(intercept)   6.46646    1.60571  4.0272 5.645e-05 ***
class.4.MB:(intercept)   3.26112    1.50658  2.1646 0.0304189 *  
class.4.price           -2.76099    1.11388 -2.4787 0.0131855 *  
(class)2                 0.58019    0.15590  3.7216 0.0001980 ***
(class)3                 0.39089    0.19172  2.0389 0.0414622 *  
(class)4                -0.20436    0.14321 -1.4270 0.1535736    
```

Demand with four-segment model



Estimated parameters of four-segment model

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

- Note that these four segments exhibit an intuitive preference dispersion.
- Three segments are loyal to each product, and there's a "bubble" segment, whose β_{0k}^{j} for bubble products is higher that β_{0k}^{KR}.

How is MLE estimating all the parameters?

- With latent segment model, we need to estimate the preference parameters of each type and the proportion of each type from the choice data alone.
- We use the same MLE, and magically multiple types of consumers are identified with different preferences. Let's do "how does gmnl estimate parameters?" exercise in this setting.
- Because this environment is much more involved than the previous ones, we will revisit an analogous, but simpler alternative - flipping coins.

Revisiting the coin-flipping example

- There are N coins on the table. We flip each coin T times, so there are $N \times T$ outcomes in the data (note the two-dimensional nature).
- Coins are uneven, and we know that there are two types of coins with different probabilities of coming up heads. We don't know the proportion of types among coins.
- We build a model that "one type of coin comes up head with probability θ_1, the other with probability θ_2". Also say "the proportion of type 1 coin is w_1".
- Using the data, can we estimate
  - θ_1 and θ_2?
  - Proportion of type 1 coins w_1?

Revisiting the coin-flipping example

- Note the analogy to the following situation:
  - We have data on N consumers' past purchase history (T times per individual).
  - There are two types of consumers with different preference parameters.
- We build a model with two types of consumers. Can we estimate
  - Preference parameters of type 1 and type 2 consumers?
  - Proportion of type 1 consumers vs type 2?

Review of MLE with single type

- First, let's review how we estimate a model without heterogeneity (all coins share the same θ) with MLE.
- Denote by $y_n = [Head_n, Tail_n]$ the outcome of flipping a single coin (n) by T times that we observe in the data. $Head_n$ is the number of heads, and $Tail_n$ is the number of tails in the data. The number of heads and tails varies across different coins (hence subscript n).
- The "likelihood that this model with a particular parameter θ explains the observed outcome of a single coin n" is

$$L(y_n \mid \theta) = \theta^{Head_n}(1-\theta)^{Tail_n}.$$

Review of MLE with single type

- Because we observe $n = 1, 2, ..., N$ coins, we need the likelihood that the model explains the outcome of *all* coins.
- Because the outcome from each coin is independent from one another, likelihood of observing outcomes of N coins is obtained by multiplying likelihood for each coin across coins.

$$L(y \mid \theta) = \prod_{n=1}^{N} L(y_n \mid \theta) = \prod_{n=1}^{N} \theta^{Head_n} (1-\theta)^{Tail_n}.$$

where I denote observations of all n by y .

Review of MLE with single type

- We can simplify the likelihood through the following operation:

$$
\begin{aligned}
L(y \mid \theta) &= \prod_{n=1}^{N} \theta^{Head_n}(1-\theta)^{Tail_n} \\
&= \theta^{Head_1}(1-\theta)^{Tail_1}\theta^{Head_2}(1-\theta)^{Tail_2}\ldots \\
&= \theta^{Head_1+Head_2+\ldots}(1-\theta)^{Tail_1+Tail_2+\ldots} \\
&= \theta^{Head}(1-\theta)^{Tail},
\end{aligned}
$$

  where "Head" is the total number of heads in the data, and "Tail" is the total number of tails in the data.
- With single type of coins, only the total number of heads and tails matters.

How is θ estimated in a single-type model?

$$L(y) = \theta^{Head}(1-\theta)^{Tail},$$

- As we discussed in Topic 2. We estimate higher θ if we see more heads than tails: if we see more heads, the first half of the expression gets more weight (larger *Head* value). So higher θ maximizes $L(y)$.
- This means, the data variation that allows us to estimate θ is *proportion of heads in the data*. We will use this variation again in the two-type case.
- Side: in fact, the θ in this case equals the proportion of heads in the data.

MLE allowing for two types of coins

- Now allow two types of coins: one with probability of coming up heads θ_1, and the other θ_2. Denote by w_1 the proportion of type 1 coins.
- We now need a "likelihood that this two-type model with parameters $\{\theta_1, \theta_2, w_1\}$, explains the observed outcome".
- Just like the single-type case, we first calculate the likelihood for each coin, $L(y_n \mid \theta_1, \theta_2, w_1)$, and then multiply it across coins to get $L(y \mid \theta_1, \theta_2, w_1)$.
- To simplify notation, I will suppress dependence on θ_1, θ_2, w_1 and just write $L(y_n)$ and $L(y)$ from now.

MLE allowing for two types of coins

- One key difference from the case of observed types (demographics): we don't know the type of each coin.
- If we know the type of each coin, calculating $L(y_n)$ is straightforward. For example, if we know that coin n is type 1, we know that it comes up head with probability θ_1. Hence,

$$L(y_n \mid \text{n type 1}) = \theta_1^{Head_n}(1-\theta_1)^{Tail_n}.$$

- Similarly, if we know that n is type 2,

$$L(y_n \mid \text{n type 2}) = \theta_2^{Head_n}(1-\theta_2)^{Tail_n}.$$

MLE allowing for two types of coins

- However, because we don't know the type of each coin, our likelihood for coin n, $L(y_n)$ needs to be an expectation over two possible types.
- The probability that a single coin is type 1 is equal to the proportion of type 1 coins, w_1. Hence the expectation is:

$$\begin{aligned} L(y_n) =& w_1 L(y_n \mid \text{n type 1}) + (1-w_1) L(y_n \mid \text{n type 2}) \\ =& w_1 \theta_1^{Head_n} (1-\theta_1)^{Tail_n} \\ &+ (1-w_1)\theta_2^{Head_n}(1-\theta_2)^{Tail_n}. \end{aligned}$$

MLE allowing for two types of coins

- Once we have $L(y_n)$, the likelihood for all coins, $L(y)$, is obtained by multiplying coin-specific likelihood across all the coins.

$$
\begin{aligned}
L(y) &= \prod_{n=1}^{N} L(y_n) \\
&= \prod_{n=1}^{N} \Big[w_1 \theta_1^{Head_n} (1-\theta_1)^{Tail_n} \\
&\qquad + (1-w_1) \theta_2^{Head_n} (1-\theta_2)^{Tail_n} \Big].
\end{aligned}
$$

- This is the likelihood that a model with two types of coins explains the observed outcome.

How are θ_1, θ_2 and w_1 estimated?

$$L(y) = \prod_{n=1}^{N} \Big[w_1 \theta_1^{Head_n} (1-\theta_1)^{Tail_n} + (1-w_1)\theta_2^{Head_n}(1-\theta_2)^{Tail_n} \Big].$$

- Intuitively, MLE of a two-segment model takes the following steps.
  1. Pick (guess) a candidate w_1.
  2. Estimate θ_1 and θ_2 associated with that w_1.
  3. We then get the likelihood associated with $\{\theta_1, \theta_2, w_1\}$.
  4. Iterate this process across many w_1 and pick the parameters that maximize the likelihood.

How are θ_1, θ_2 and w_1 estimated?

- From here, I assume that $\theta_1 > \theta_2$ without loss of generality.
- To begin the process, make a guess about the value of w_1 - how many coins are type 1. Let's say we guess $w_1 = 0.8$.
- This is equivalent to assuming that "80% of coins are more likely to come up head than the other 20%".

How are θ_1, θ_2 and w_1 estimated?

- We observe outcomes from all the coins in the data. Some coins come up head more often (large $Head_n$) than others. Let's sort all the coins according to the number of heads.
- Then our guess of $w_1 = 0.8$ implies that the top 80% of observations are more likely to be type 1 coins, and the bottom 20% of the coins are more likely to be type 2.
- In practice, we are never sure which coin is which type (some type 2 coins may happen to produce many $Head_n$ by chance). But for simplicity let's believe for now that top 80% is type 1.

How are θ_1, θ_2 and w_1 estimated?

$$L(y) = \prod_{n=1}^{N} \Big[w_1 \theta_1^{Head_n} (1 - \theta_1)^{Tail_n} \\ + (1 - w_1) \theta_2^{Head_n} (1 - \theta_2)^{Tail_n} \Big].$$

- Let's estimate θ_1. θ_1 is estimated based on the proportion of heads *among all type 1 coins* (analogous to the estimation of θ in no-segmentation case). i.e. the proportion of heads from top 80% of the coins.

How are θ_1, θ_2 and w_1 estimated?

$$L(y) = \prod_{n=1}^{N} \Big[w_1 \theta_1^{Head_n} (1-\theta_1)^{Tail_n} + (1-w_1) \theta_2^{Head_n} (1-\theta_2)^{Tail_n} \Big].$$

- Similarly, we can estimate θ_2 based on the proportion of heads of bottom 20% of the coins.

How are θ_1, θ_2 and w_1 estimated?

$$L(y) = \prod_{n=1}^{N} \Big[w_1 \theta_1^{Head_n} (1-\theta_1)^{Tail_n} + (1-w_1)\theta_2^{Head_n}(1-\theta_2)^{Tail_n} \Big].$$

- Because we have $\{\theta_1, \theta_2, w_1\}$, we can evaluate the likelihood. At different guess of w_1, we can do the same.
- Eventually, we can pick the values of w_1 such that along with the associated θ_1 and θ_2, it maximizes the likelihood.

How are θ_1, θ_2 and w_1 estimated?

- When latent segments are involved, estimation of the model becomes much less intuitive. There's no clear-cut explanation that "this variation in the data estimates this parameter" anymore.
- Nevertheless, we can still see that
  1. Proportion of heads among the coins that has a high $Head_n$ estimates θ_1.
  2. Proportion of heads among the low $Head_n$-ones estimates θ_2.
  3. The proportion of high $Head_n$-ones vs low $Head_n$-ones estimates w_1.

One caveat

- As we said, we are never sure about which coin is which type. Hence this "Top 80% estimates θ_1, bottom 20% estimates θ_2" example is a heuristic statement.
- In practice, we use all observations to estimate θ_1, but coins with higher $Head_n$ gets a higher weight.
- Similarly, we use all observations to estimate θ_2 with low $Head_n$ coins get more weight.

An analogy to choice data

- Although more complicated, the estimation mechanism in the context of choice data is analogous.
  - Proportion of purchases among those who buy KB most often estimates β^j_{0KB} and β_{1KB}.
  - Proportion of purchases among those who buy KR most often estimates β^j_{0KR} and β_{1KR}.
  - and so on...

Predicting demand using the estimated model

- Now let's turn to predicting demands using the estimated parameters.
- In fact, demand prediction can be done in the exact same way as in K-mean clustering case. This is because in both cases, we are predicting demand of K discrete segments.
- The only difference is "how we group consumers together". Once they are grouped, everything else is identical.

Repost: estimates from K-mean clustering

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
1	1	6.752798	8.469717	7.788226	-6.105475
2	2	1.449974	1.905069	0.981285	-2.004867
3	3	8.353175	6.550447	7.429280	-6.358137
4	4	5.945230	6.012463	5.771539	-4.555660
5	5	6.229873	5.886986	6.184821	-5.792098
6	6	11.219509	10.772875	10.324222	-7.970750

```
> seg.share
   1    2    3    4    5
0.12 0.13 0.28 0.19 0.14 0.14
```

- With K-means approach, we had K distinct segments of consumers. For each segment, we had β_{0k}^{j}, β_{1k} and the proportion of that type, w_k.
- The structure of these parameters is exactly identical to latent-type models.

Estimates from four-segment model

```
Coefficients:
                        Estimate Std. Error z-value  Pr(>|z|)
class.1.KB:(intercept)  4.57734    1.68285  2.7200 0.0065284 **
class.1.KR:(intercept)  2.30125    1.71091  1.3450 0.1786098
class.1.MB:(intercept)  4.88573    2.17868  2.2425 0.0249277 *
class.1.price          -5.52225    1.67580 -3.2953 0.0009832 ***
class.2.KB:(intercept)  3.86201    1.31516  2.9365 0.0033190 **
class.2.KR:(intercept)  4.81445    0.89045  5.4068 6.417e-08 ***
class.2.MB:(intercept)  6.13200    0.91236  6.7210 1.805e-11 ***
class.2.price          -4.69838    0.68504 -6.8585 6.958e-12 ***
class.3.KB:(intercept)  6.22360    1.18477  5.2530 1.496e-07 ***
class.3.KR:(intercept)  4.28802    1.14026  3.7605 0.0001695 ***
class.3.MB:(intercept)  4.99226    1.24903  3.9969 6.418e-05 ***
class.3.price          -4.11882    0.98987 -4.1610 3.169e-05 ***
class.4.KB:(intercept)  3.88780    1.56668  2.4815 0.0130813 *
class.4.KR:(intercept)  6.46646    1.60571  4.0272 5.645e-05 ***
class.4.MB:(intercept)  3.26112    1.50658  2.1646 0.0304189 *
class.4.price          -2.76099    1.11388 -2.4787 0.0131855 *
(class)2                0.58019    0.15590  3.7216 0.0001980 ***
(class)3                0.39089    0.19172  2.0389 0.0414622 *
(class)4               -0.20436    0.14321 -1.4270 0.1535736
```

Single-price optimization with latent segment models

- If we set a single price (no targeting), the optimal pricing problem is also identical to the Kmeans case.
- We first calculate the aggregate choice probability, which is the weighted average of all types' choice probabilities. We then multiply it with the number of consumers in the market, obtain the aggregate demand and maximize the profit against it.
- Because these aspects aren't new, let's skip them for now and move on to something new - targeting.

Targeted pricing with latent segment models

- In order to engage in targeting, we need to know the choice probability of each consumer n, which differs across consumers. By now, we have estimated the choice probability of each segment k of consumers. What remains is to know who (n) belongs to which segment (k).
- However, with latent segments of consumers, we don't know who belongs to which segment - this is an extra layer of difficulty compared to the case of demographic-based segmentation.
- In order to calculate choice probability of each consumer, we need to *predict* which segment each consumer belongs to.

Predicting each consumer's type

- Let's start from a simple prediction. Consider a situation that we estimated a model with two latent segments, and $w_1 = 0.8$. The proportion of type 1 consumer is 80%.
- Then we can say that with probability 0.8, each consumer in the data is type 1.
- But this doesn't help us much - because w_1 is common across all consumers, *any* consumer in the data is type 1 with probability 0.8. In this sense, the predicted choice probability is identical across everyone - we still cannot target anyone.
- Any way to provide a better *personalized* prediction?

Calculating posterior probability

- From here, assume that we estimated a model with two latent segments.
- We observe each consumer's history of purchases. Some consumers' behaviors are closer to the predicted actions of type 1 consumers in the model. Others' behaviors are closer to those of type 2.
- If a consumer seems to be behaving like type 1, we may say that she is more likely to be type 1. i.e. "the probability that that person is type 1" $> w_1$.
- Hence, we may update our prediction about each consumer's type based on her observed actions.

Calculating posterior probability

- Mathematically, "using the observed actions to better predict each consumer's latent type" is achieved by calculating *posterior probability*.
- w_1 that we estimated earlier corresponds to the probability that *anyone* in this market is type 1. This is called *prior probability* and is usually denoted by $Pr(n \text{ is type } 1)$.
- Having observed each consumer n's history of purchases, we can predict better the probability that *she* is type 1 - which is $Pr(n \text{ is type } 1 | \, y_n)$, where $y_n = \{y_{n1}, ..., y_{nT}\}$ represents the history of choices n made during the sample period.
- This "probability of latent type conditional on the observed actions" is called *posterior probability*.

Bayes rule

- We can calculate the posterior probability by "Bayes Rule", which takes the following form.

$$Pr(n \text{ is type } k \mid y_n) = \frac{Pr(n \text{ is type } k \text{ and } y_n)}{\sum_{k'=1}^{K} Pr(n \text{ is type } k' \text{ and } y_n)}.$$

- Mathematically, the posterior probability is defined by the ratio of two joint probabilities. $Pr(n \text{ is type } k \text{ and } y_n)$ is the probability that consumer n is type k *and* she takes actions y_n.

Bayes rule: intuition

$$Pr(n \text{ is type } k \mid y_n) = \frac{Pr(n \text{ is type } k \text{ and } y_n)}{\sum_{k'=1}^{K} Pr(n \text{ is type } k' \text{ and } y_n)}.$$

- If we write it down using plain words, the intuition becomes clear.
- The denominator is "the probability that consumer n chooses y_n at all". It could be that "n is type 1 and choose y_n", or "n is type 2 and choose y_n", etc. The probability that we see y_n at all is the sum over all possibilities.
- The numerator is "what is the probability that consumer n is type k and choose y_n", which is one of the elements of the denominator.

Bayes rule: intuition

$$Pr(n \text{ is type } k \mid y_n) = \frac{Pr(n \text{ is type } k \text{ and } y_n)}{\sum_{k'=1}^{K} Pr(n \text{ is type } k' \text{ and } y_n)}.$$

- By taking the ratio between the two, we calculate "how likely a type k consumer chooses y_n *relative to* other type k' consumers".
- Suppose that we see someone taking action y_n, and that the model predicts that type 1 consumers love that option. Then $Pr(n \text{ is type } 1 \text{ and } y_n)$ is much higher than that of any other type k'. Hence the ratio is higher.
- Higher ratio implies that *having seen* y_n, we assign a higher probability that n is type 1. i.e. because n takes the option that type 1 people love, we conclude that n is more likely to be type 1.

Example of posterior probability calculation

- Consider a following example.
  1. The proportion of type 1 consumers (= prior probability that each consumer is type 1) is 0.8 (w_1).
  2. The probability that a type 1 consumer chooses the actions y_n is 0.5 (the conditional likelihood - slide 27).
  3. The probability that a type 2 consumer chooses the actions y_n is 0.2.
- If we see consumer n choosing y_n in the data, what is the posterior probability that n is type 1?

Example of posterior probability calculation

- First we calculate the numerator - the probability that n is type 1 *and* she chooses y_n.

$$
\begin{aligned}
&Pr(n \text{ is type 1 and } y_n)\\
&= \text{Prior probability that } n \text{ is type 1}\\
&\quad \times \ \text{Probability that type 1 consumer chooses } y_n\\
&=0.8 \times 0.5\\
&=0.4.
\end{aligned}
$$

Example of posterior probability calculation

- Next we calculate the denominator. We first calculate $Pr(n \text{ is type 2 and } y_n)$, and add the number to $Pr(n \text{ is type 1 and } y_n)$ calculated earlier.

$$
\begin{aligned}
& Pr(n \text{ is type 2 and } y_n) \\
& = \text{Prior probability that } n \text{ is type 2} \\
& \quad \times \; \text{Probability that type 2 consumer chooses } y_n \\
& = 0.2 \times 0.2 \\
& = 0.04.
\end{aligned}
$$

- Hence the denominator is $0.4 + 0.04 = 0.44$.

Example of posterior probability calculation

- Hence, the posterior probability that consumer n is type 1 is:

$$Pr(n \text{ is type } 1 \mid y_n) = \frac{0.4}{0.44} = 0.91.$$

- Having observed y_n, we assign a higher probability that n is type 1 (as opposed to the original 0.8).
- Following the same procedure, we can calculate the posterior probability that each consumer is type k. Using the observations of past purchases, we can predict the type of each consumer better.

So, where are we now?

- Our goal is targeting - to this end, we need to predict β^j_{0n} and β_{1n} for each consumer n.
- We know β^j_{0k} and β_{1k} for each segment k, but we didn't know who (n) belongs to which segment (k).
- We just developed the posterior probability $Pr(n \text{ is type } k \mid y_n)$. We can now associate n with k.
- Let's compute β^j_{0n} and β_{1n} next.

Predict preferences of each consumer n

- Once we know the posterior probability for each consumer n, we can now predict β_{0n}^j and β_{1n} as follows.

$$\beta_{0n}^j = \sum_{k=1}^{K} Pr(n \text{ is type } k \mid y_n)\beta_{0k}^j,$$
$$\beta_{1n} = \sum_{k=1}^{K} Pr(n \text{ is type } k \mid y_n)\beta_{1k}.$$

- Predicted preference location of each consumer n is a weighted average of type k consumer's preference; We know the preference of each type k. We also know the probability that n belongs to k. Hence our prediction is simply an expectation.

Predict choice probability of each consumer n

- If we know each consumer's β_{0n}^{j} and β_{1n}, we can predict that consumer's choice probability using the logit formula.

$$\tilde{Pr}_n(y = j \mid P) = \frac{\exp(\beta_{0n}^{j} + \beta_{1n} P^{j})}{1 + \sum_{j'=1}^{J} \exp(\beta_{0n}^{j'} + \beta_{1n} P^{j'})},$$

  where β_{0n}^{j} and β_{1n} are the ones obtained in the previous slide.

- This is an *individual-level* prediction of consumer decisions, based on which we can engage in targeting.

Side: Predict preferences of each consumer n without y_n

- If we don't observe a consumer's y_n, what is our prediction of that consumer's β_{0n}^j and β_{1n}?
- It is a weighted average with the prior probability:

$$\beta_{0n}^j = \sum_{k=1}^{K} w_k \beta_{0k}^j, \qquad \beta_{1n} = \sum_{k=1}^{K} w_k \beta_{1k}.$$

- The idea is the same as before - we take expectation over β_{0k}^j and β_{1k}. If we don't know anything about n, our best guess about the probability that person n belongs to type k is our prior, w_k.
- This is not a personalized prediction - nowhere in the expression do we use information about person n (because we don't have any).

Calculate β_{0n}^{j} and β_{1n} in R

- Let's calculate β_{0n}^{j} and β_{1n} in R. We don't need to hand-code Bayes rule - "effect.gmnl" function does it for us. We plug in the outcome of gmnl ("lc4" in the lecture code).
- It returns two objects - "mean" and "sd.est". The predicted preference parameters are stored in "mean".

```
#Calculate each consumer's predicted beta0 and beta1 using posterior.
aux=effect.gmnl(lc4)
postcoef=aux$mean
```

```
     KB:(intercept) KR:(intercept) MB:(intercept)     price
[1,]       4.596297       3.815527       5.380521 -4.845652
[2,]       5.000851       4.559740       5.582005 -4.419240
[3,]       3.887797       6.466461       3.261122 -2.760993
[4,]       4.516809       2.665130       5.048998 -5.388303
[5,]       4.412077       3.207927       5.288289 -5.189891
[6,]       4.202951       4.676461       5.043721 -4.641030
```

- Each row corresponds to the predicted parameters, β_{0n}^{j} and β_{1n}, of each consumer n, evaluated based on the posterior probability.

Calculate β_{0n}^{j} and β_{1n} in R

- Let's see if it is working in a way we expect. Here are the estimated parameters for each segment, β_{0k}^{j} and β_{1k}.

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

- Consider a consumer with the following history of purchases:

```
[1] "KR" "KR" "KR" "KR" "KR" "KB" "KR" "KR" "KR" "KR" "0"
    "MB" "KR" "KR" "KR" "KR" "KR" "KR" "KR" "MB" "MB" "MB"
```

- "effect.gmnl" calculate $Pr(n \text{ is type } k \mid y_n)$ based on this observation and calculate β_{0n}^{j} and β_{1n} as the weighted average of β_{0k}^{j} and β_{1k}. What are the β_{0n}^{j} and β_{1n} we expect to see?

Calculate β_{0n}^{j} and β_{1n} in R

```
             segment intercept.KB intercept.KR intercept.MB price.coef
Bubble type        1     4.577341     2.301254     4.885727  -5.522249
MB type            2     3.862012     4.814451     6.131997  -4.698380
KB type            3     6.223598     4.288017     4.992259  -4.118821
KR type            4     3.887797     6.466461     3.261122  -2.760993
```

```
 [1] "KR" "KR" "KR" "KR" "KR" "KB" "KR" "KR" "KR" "KR" "0"
     "MB" "KR" "KR" "KR" "KR" "KR" "KR" "KR" "MB" "MB" "MB"
```

- We get these values:

```
KB:(intercept) KR:(intercept) MB:(intercept)          price
      3.887797       6.466461       3.261122      -2.760993
```

- A lot of "KR" observation is sufficient to conclude that this person is KR type (by construction of Bayes rule, we are never sure - in this case, we are 99.999% sure).

Calculate β^j_{0n} and β_{1n} in R

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

- What about this person?

```
"KB" "0"  "0"  "KB" "KB" "0"  "0"  "0"  "0"  "0"
"0"  "KB" "0"  "0"  "0"  "KB" "KB" "KR" "0"  "0"
```

Calculate β_{0n}^{j} and β_{1n} in R

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

"KB" "0" "0" "KB" "KB" "0" "0" "0" "0" "0"

"0" "KB" "0" "0" "0" "KB" "KB" "KR" "0" "0"

- We get these values:

KB:(intercept)	KR:(intercept)	MB:(intercept)	price
5.989571	4.006867	4.977605	-4.317784

- "KB" observations lead us to believe that he may be KB type. However, with lot of zeros, we are not that sure yet - hence the predicted parameters are not exactly those of KB type.

Calculate β_{0n}^{j} and β_{1n} in R

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

- What about this person?

"0" "0" "0" "0"

Calculate β_{0n}^{j} and β_{1n} in R

	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

"0" "0" "0" "0"

- We get these values.

KB:(intercept)	KR:(intercept)	MB:(intercept)	price
4.516809	2.665130	5.048998	-5.388303

- It is tempting to say that this purchase history shouldn't provide any information about the type of this consumer.
- If that's the case, these values should equal to β_{0n}^{j} and β_{1n} calculated based on the prior probability (e.g., $\beta_{0n}^{j} = \sum_k w_k \beta_{0k}^{j}$).

Calculate β_{0n}^j and β_{1n} in R

- In practice, there's a significant difference between what we get (first line) and the predicted parameters based on the prior (second line).

	KB:(intercept)	KR:(intercept)	MB:(intercept)	price
effect.gmnl outcome	4.516809	2.665130	5.048998	-5.388303
Prior-based weighted avg	4.694214	4.431615	5.094290	-4.381010

- This indicates that her purchase history does provide information about her type, hence $Pr(n \text{ is type } k \mid y_n) \neq w_k$.
- But how?

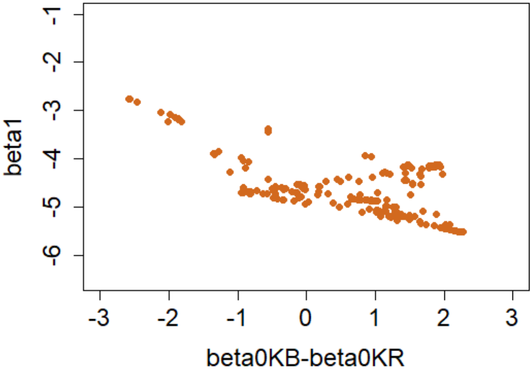
Calculate β_{0n}^{j} and β_{1n} in R

KB:(intercept)	KR:(intercept)	MB:(intercept)	price
4.516809	2.665130	5.048998	-5.388303

- "Not having bought any" indicates that this consumer is price sensitive. Hence she is more likely to be bubble type, and less likely to be KR type. Hence we update her parameters as such.
- Note that her price coefficient is much closer to that of bubble type.

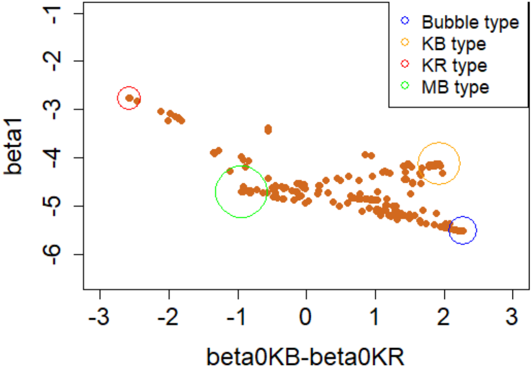
	segment	intercept.KB	intercept.KR	intercept.MB	price.coef
Bubble type	1	4.577341	2.301254	4.885727	-5.522249
MB type	2	3.862012	4.814451	6.131997	-4.698380
KB type	3	6.223598	4.288017	4.992259	-4.118821
KR type	4	3.887797	6.466461	3.261122	-2.760993

Examine the estimated β_{0n}^{j} and β_{1n}



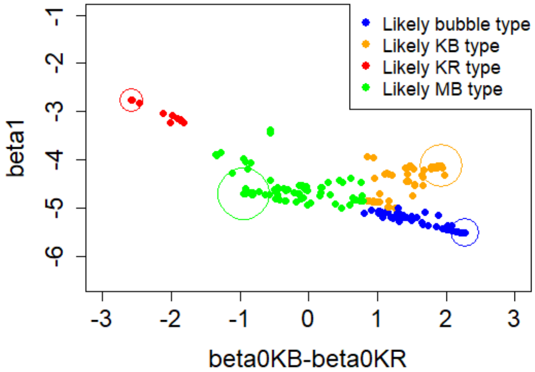
- Once we have β_{0n}^{j} and β_{1n}, we can draw a scatterplot, similar to the one we had before.
- Each dot is one consumer - those who prefer KB are more price sensitive.

Examine the estimated β_{0n}^{j} and β_{1n}



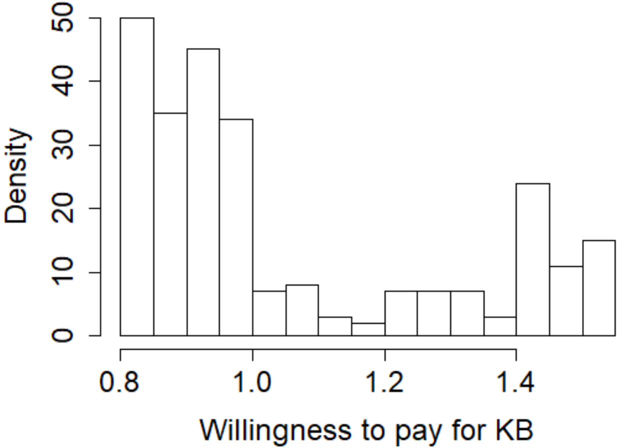
- Overlaying with the location of β_{0k}^{j} and β_{1k} of each segment - each consumer's β_{0n}^{j} and β_{1n} is located in between the β_{0k}^{j} and β_{1k}.
- This is natural, because expectation is nothing but a weighted average between β_{0k}^{j}.

Examine the estimated β_{0n}^{j} and β_{1n}



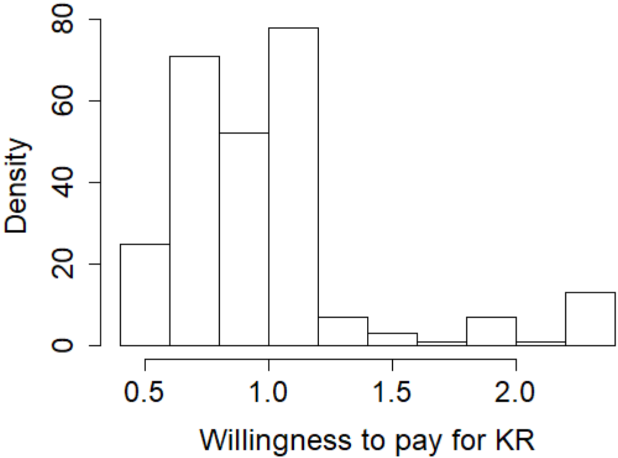
- Let's color each consumer differently - red is "likely KR type", orange is "likely KB type", etc. We can then offer targeted pricing to different segment of consumers.

Distribution of WTP for KB



- We can calculate WTP of each consumer by $-\frac{\beta^j_{0n}}{\beta_{1n}}$.
- Distribution of WTP for KB - "likely KB type" people on the right.

Distribution of WTP for KR



- Distribution of WTP for KR - "likely KR type" people on the right. Note that KR is a much smaller segment than KB or bubble segment.

Latent type or demographics?

- When should we use latent type models vs demographic-based models?
- In general, latent type models are superior than demographic-based models. While demographic variables are not always a good predictor of the underlying preferences, latent type models find consumer segments based on each consumer's revealed preference directly.
- There is one disadvantage for latent type models: it requires a lot of observations of purchase history for each consumer.

Latent type or demographics?

- Recall that latent segment models find segmentation based on consumers' past purchase history - for this approach to be reliable, we need a lot of observations *per consumer*.
- Segmentation is estimated based on how consumers *persistently* behave differently from one another. Some *persistently* choose KR whereas others *persistently* choose KB, etc. Then we know there are KR lovers and KB lovers. If no persistence exists, we cannot find segments.
- These constraints preclude the use of latent type models in certain industries, e.g. durable goods (cars, houses, etc), or when you are new to the market.

Latent type or demographics?

- Even if the latent-type models are estimable, still one problem, in the context of targeting.
- For anyone we don't see in the data, we have no predictions about her segment - we assign β_{0n}^{j} and β_{1n} based on the prior, which is identical across all consumers.
- This is unlike the case of demographic-based segmentation, where you can predict each consumers' preference from the observations of *other consumers* with the same demographic background.

Summary

- We studied models with latent consumer types. We considered implementation in R, intuition on the estimation algorithm and the way to apply the model to optimal uniform/targeted pricing.
- Models of latent segments are a driving horse of demand prediction in many industries. Many firms operating for long in the market have access to long history of consumers' purchase record. Segmenting and positioning/targeting based on latent types is an appealing option.
- On the other hand, if your firm is new in the market, you unlikely have access to long enough panel of choice data, hence the gains from estimating latent type models may be limited.