Importance Sampling

Jan Overgoor

Contents


```
to build from command-line, run with:
R -e "rmarkdown::render('sampling.Rmd', output_file='sampling.html')"
```

First we read the data.

Data is constructed as follows:

- start with `G = nx.erdos_renyi_graph(2000, 0.005, seed=None, directed=False)`
- construct $n = 1000$ edges, sample i uniformly, sample j according to conditional logit with $u_{ij} = 0.5 \cdot \log k_j + 2 \cdot 1\{FoF_{ij}\}$.

1. Single Models

Model 1 - No Sampling

Does it model utility well? We fit $p(x_{ij}) \sim \theta_0 \cdot u_{ij}$. The ground truth is $\theta_0 = 1$.

```
##        u
## 1.006169
```

Does it fit the parameters well? We fit $p(x_{ij}) \sim \theta_1 \cdot k_j + \theta_2 \cdot fof_{ij}$. The ground truth is $\theta_1 = 0.5$ and $\theta_2 = 2$.

```
##  log(deg)       fof
## 0.4129059 2.0320689
```

It slighly underestimates θ_1, the role of degree.

Model 2 - uniform sampling (s=10)

Does it model utility well?

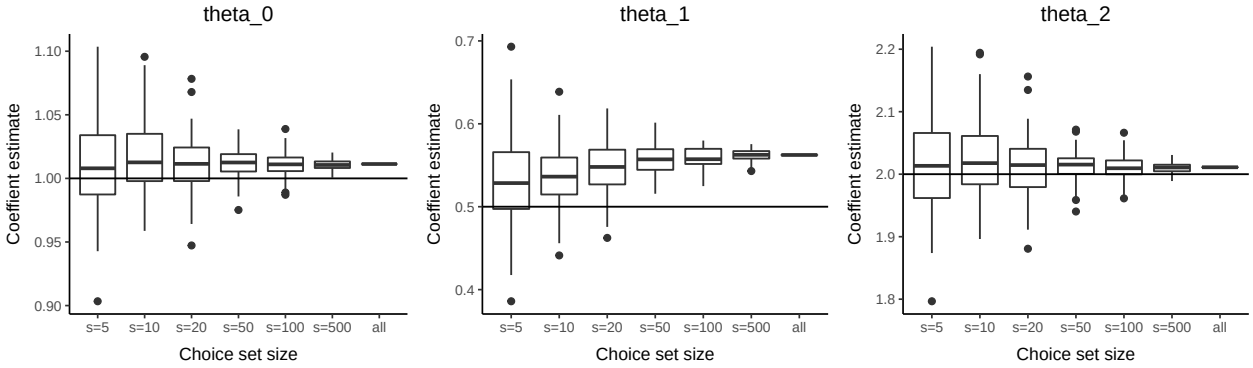
```
##         u
## 0.9889336
```

Does it fit the parameters well?

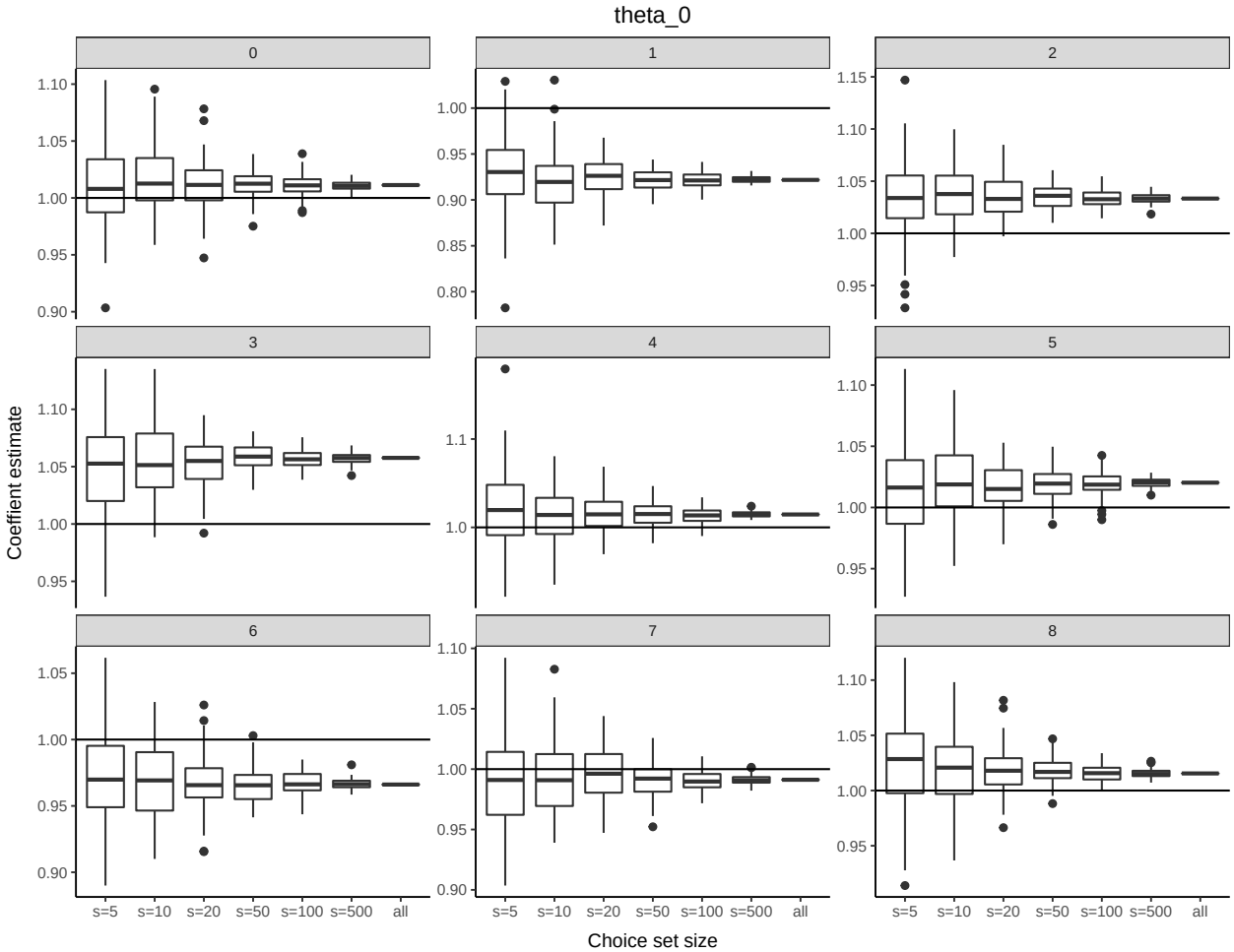
```
##  log(deg)       fof
## 0.4385999 1.9917675
```

2. Many models

In Python, for $s \in [5, 10, 20, 50, 100]$, do uniform sampling 100 times, compute both utility model and parameter model.

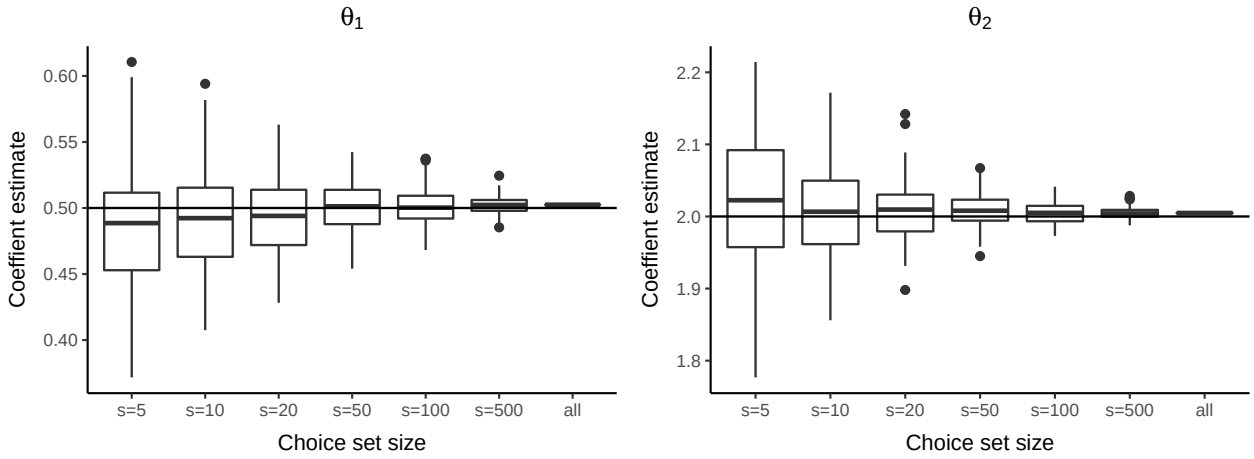


The estimates converge to `all`, with less variance as s decreases. However, there is a bias in the parameter estimates, even for `all`.



This looks to be variance in the data generating process.

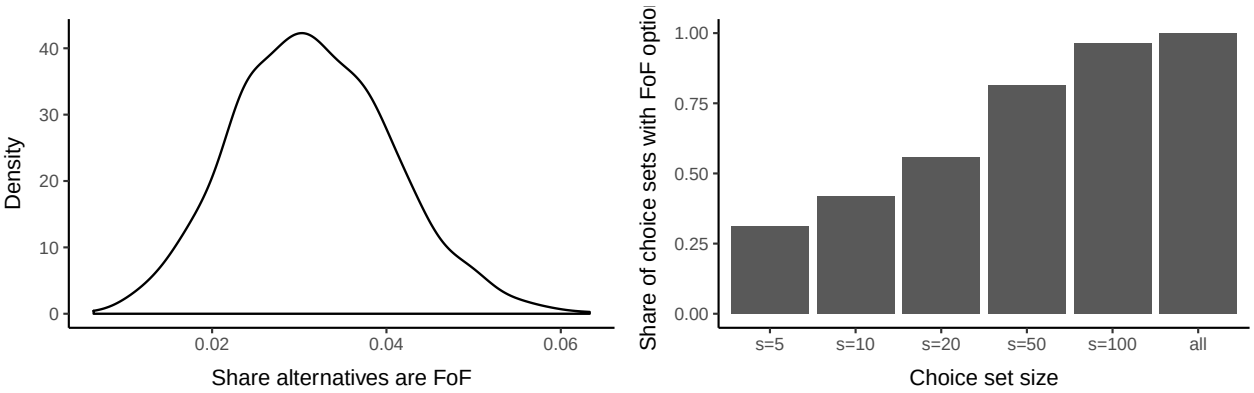
Here is a version with better presentation:



3. Non-Uniform sampling

$u_{=}0.5 \cdot k_x + 2 \cdot FoF_x$

What share of alternatives has FoF option?

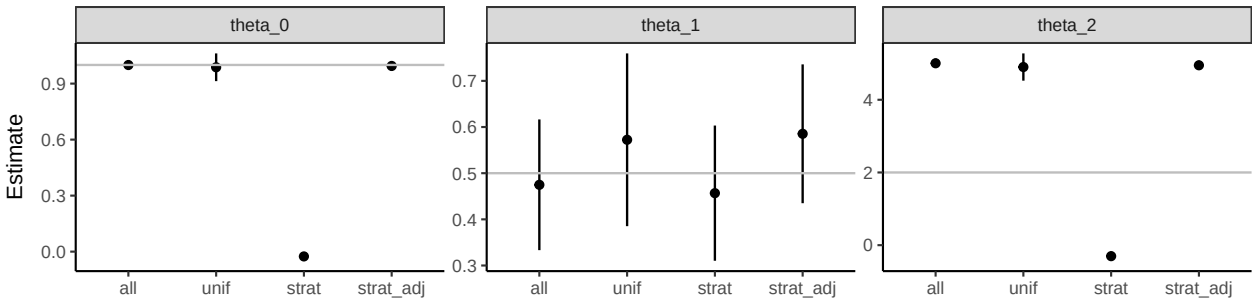


Fit with uniform sampling and stratified by Fof (both $s = 10$).

```
##
## ============================================================
##                              Dependent variable:
##                  ------------------------------------------
##                                       y
##                     Unif       Strat        Unif       Strat
##                     (1)         (2)         (3)         (4)
## ------------------------------------------------------------
## u                 1.039***   -0.580***
##                   (0.050)     (0.038)
## log(deg)                                  0.579***    0.584***
##                                           (0.121)     (0.121)
## fof                                       2.061***   -1.333***
##                                           (0.105)     (0.081)
## ------------------------------------------------------------
## Observations        965         965         965         965
## Log Likelihood -2,122.181 -2,173.452 -2,122.055 -2,145.290
```

```
## ============================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

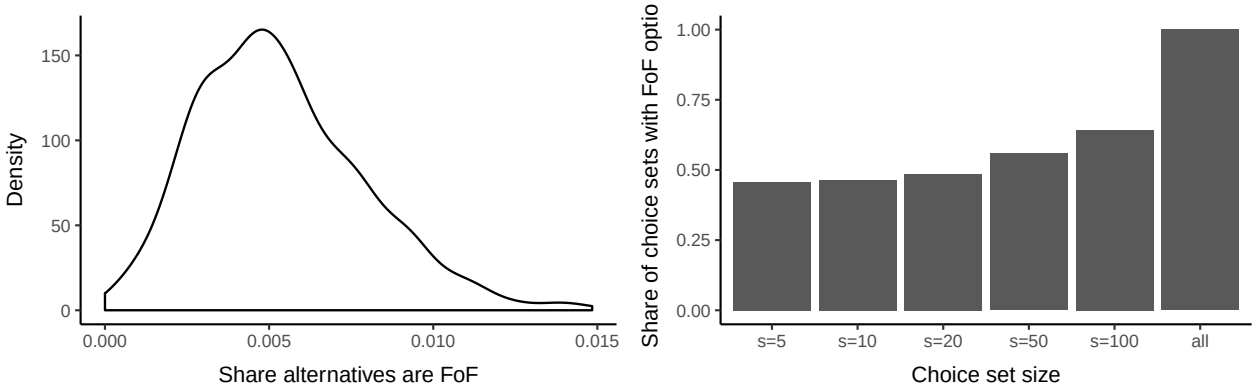
Now, result from doing it correctly (fitting in Python).



Method	Covariate	Mean	SE
all	theta_ 0	0.9995887	0.0128838
unif	theta_ 0	0.9875032	0.0378979
strat_adj	theta_ 0	0.9945701	0.0128677
all	theta_ 1	0.4748447	0.0722481
unif	theta_ 1	0.5723740	0.0954774
strat_adj	theta_ 1	0.5853502	0.0767333
all	theta_ 2	5.0041964	0.0657916
unif	theta_ 2	4.9005008	0.1913626
strat_adj	theta_ 2	4.9489058	0.0659108

$u_{=}0.5 \cdot k_x + 5 \cdot FoF_x$

What share of alternatives has FoF option?

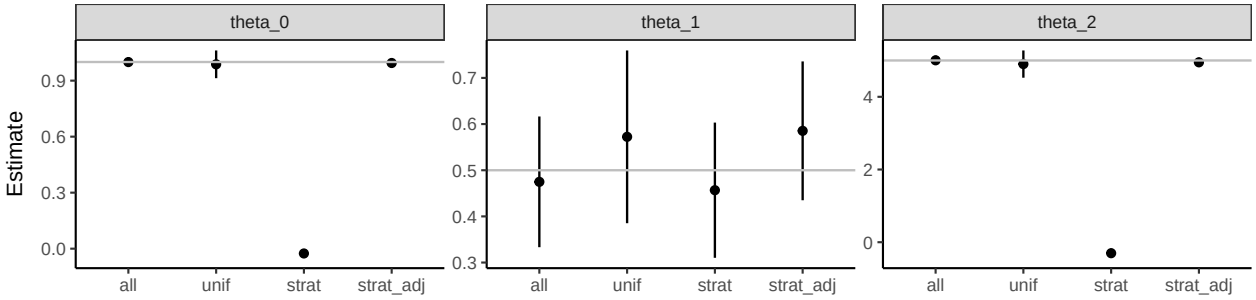


Fit with uniform sampling and stratified by Fof (both $s = 10$).

```
##
## ============================================================
##                              Dependent variable:
##                    ----------------------------------------
##                                       y
##                       Unif       Strat       Unif       Strat
##                       (1)         (2)        (3)         (4)
## ------------------------------------------------------------
## u                   1.022***   -0.027**
```

```
##                    (0.041)    (0.012)
## log(deg + 1e-08)                          0.547***   0.475***
##                                           (0.094)    (0.077)
## fof                                       5.089***  -0.299***
##                                           (0.210)    (0.065)
## ------------------------------------------------------------
## Observations       1,000      1,000      1,000      1,000
## Log Likelihood   -1,426.607 -2,392.948 -1,426.529 -2,366.562
## ============================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

Now, result from doing it correctly (fitting in Python).



Method	Covariate	Mean	SE
all	theta_ 0	0.9995887	0.0128838
unif	theta_ 0	0.9875032	0.0378979
strat_adj	theta_ 0	0.9945701	0.0128677
all	theta_ 1	0.4748447	0.0722481
unif	theta_ 1	0.5723740	0.0954774
strat_adj	theta_ 1	0.5853502	0.0767333
all	theta_ 2	5.0041964	0.0657916
unif	theta_ 2	4.9005008	0.1913626
strat_adj	theta_ 2	4.9489058	0.0659108