

Estimating the following model:

$$f(b, t) = Stations * Pop + \alpha Corp * Pop + \beta Dist + \epsilon$$

To examine the impact of corporate ownership and distance on radio station mergers.

Following Akkus, Cookson, and Hortaçsu (2016) using the maximum score estimator:

$$\beta = \operatorname{argmax}_{\beta} Q(\beta) = \sum_{y=1}^2 \sum_{b=1}^{M_y-1} \sum_{b'=b+1}^{M_y} 1[f(b, t|\beta) + f(b', t'|\beta) \geq f(b', t|\beta) + f(b, t'|\beta)]$$

To do so, I create a two payoff matrices one for actual matches and counter-factual matches.

The payoff matrix for actual matches is simply an $M \times 1$ vector where M is the number of buyer/target matches in a given year. The counter-factual pay-off matrix is $M \times (M-1)$ where the elements are as below:

$$(i, j) \text{ element} = \begin{cases} f(\text{buyer } i, \text{target } (j+1)), & \text{if } j < (i - 1) \text{ or } i = M \\ f(\text{buyer } i, \text{target } (j + 2)), & \text{otherwise} \end{cases}$$

I use list comprehension in python (due to its speed relative to standard for loops) to create and evaluate these matrices during the optimization process (see `max_score.py` and `payoff_func.py`)

After transforming price and population into hundreds of thousands of people and dollars, I receive the following results from this estimation (See Results 1 section below):

$$f(b, t) = Corp * Pop + 274.77 * Stations * Pop + -2.33 * Dist + \epsilon$$

using Differential Evolution method, and

$$f(b, t) = Corp * Pop + 277.5 * Stations * Pop + -2.3 * Dist + \epsilon$$

using Nelder-Mead method.

Note that the Nelder-Mead is particularly sensitive to the initial guess. The initial guess I used is $\alpha = 300$ and $\beta = -2$. These guesses were derived from results of the differential evolution method using fairly wide bounds.

The differential evolution method is quite inconsistent here, especially in the estimation of alpha (see Results 1a below). I contend this is because the variable Corp_ Owner_ Buyer has very little variation. In fact, only 2 buyers across all years have a non-zero value for this variable. Therefore, α has very little impact on the value of the equation, leading the optimizer to these shaky results.

As such, I hesitate to give much interpretative power to the magnitude of this estimate. Its sign does appear to be consistently positive (even when extending the bounds into the negative quadrant) so I am comfortable stating that corporate ownership appears to increase the desired post merger size (stations * population) of a potential merger.

The coefficient estimated for Distance is much more consistent, even when extending its bounds. It appears that an increase in distance of 1 mile decreases the desired post merger size by about 200,000.

To identify target specific characteristics, we must use a slightly different estimator. To estimate the following equation:

$$f(b, t) = \delta Stations * Pop + \alpha Corp * Pop + \gamma HHI + \beta Dist + \epsilon$$

we use the following:

$$\beta = \operatorname{argmax}_{\beta} Q(\beta) = \sum_{y=1}^2 \sum_{b=1}^{M_y-1} \sum_{b'=b+1}^{M_y} 1[f(b, t|\beta) - f(b, t'|\beta) \geq p_{bt} - p_{b't'} \ \& \ f(b', t'|\beta) - f(b', t|\beta) \geq p_{b't'} - p_{bt}]$$

In doing so, I find the following results:

$$f(b, t) = 619.31 * Stations * Pop + 149.46 * Corp * Pop - 57.60 * HHI - 446.8 * Dist + \epsilon$$

I again find some consistency issues here, refer to my above comments regarding the Corp variable. Taking these estimates at face value, we again find that corporate ownership increases the desired post merger size and distance reduces it. The new coefficient here is γ , which illustrates that a higher degree of concentration in the target market reduces the desired post merger size.

The coefficient estimates on Distance is not incredibly different (after normalization of δ) from the previous specification. However, the coefficient on α is very different in absolute and relative magnitude. As noted above, these results seem to be inconsistently estimating this coefficient due to the lack of variation in Corp variable.

See results 2 section for estimation output.

Results 1.

```
final_simplex: (array([[ 277.5          ,  -2.3          ],
 [ 277.49994278,  -2.3          ],
 [ 277.49995708,  -2.30000019]]), array([-2280., -2280., -2280.]))
    fun: -2280.0
    message: 'Optimization terminated successfully.'
     nfev: 84
      nit: 24
    status: 0
    success: True
         x: array([ 277.5,  -2.3])
    fun: -2280.0
    message: 'Optimization terminated successfully.'
     nfev: 123
      nit: 3
    success: True
         x: array([ 274.77620508,  -2.33053932])
```

Results 1a.

Below we see the results from 4 consecutive estimations. Note that the minimized function value does not change, but the estimate of alpha changes greatly.

```
In [2]: run PS4
      fun: -2279.0
      message: 'Optimization terminated successfully'
      nfev: 153
      nit: 4
      success: True
      x: array([ 386.61594798,  -2.59404167])
```

```
In [4]: run PS4
      fun: -2280.0
      message: 'Optimization terminated successfully'
      nfev: 123
      nit: 3
      success: True
      x: array([ 275.31252328,  -2.34465678])
```

```
In [6]: run PS4
      fun: -2280.0
      message: 'Optimization terminated successfully'
      nfev: 123
      nit: 3
      success: True
      x: array([ 337.65761796,  -2.3687425 ])
```

```
In [8]: run PS4
      fun: -2280.0
      message: 'Optimization terminated successfully'
      nfev: 183
      nit: 5
      success: True
      x: array([ 190.56822806,  -2.2963926 ])
```

Results 2.

```
In [10]: run PS4.py
         fun: -1805.0
         message: 'Optimization terminated successfully.'
         nfev: 185
         nit: 2
         success: True
         x: array([ 619.31288005,  149.45797186, -57.60658558, -446.08013467])
```