Assortative Matching with Coarse Types: Estimation of Parameters

Michael Peters, James Yuming Yu

Vancouver School of Economics

December 3, 2021

Recap: Setup

- set D of academic departments (buy and sell)
- set S of sinks (govt, private sector; buy only)
  - $n_d + n_s = \sum_{j \in D \cup S} n_j = n$
- each $d_t \in D$ has a tier/type t and value v_t such that $v_t > v_{t+1}$
  - $t \in 1..k$, k fixed
  - $D_t = \{d_j \in D : j = t\}$
  - $n_t = \sum_{j \in D_t} n_j$ and $n_s = \sum_{j \in S} n_j$

Recap: Setup

- each department of type t has offers $x_t \sim F_t[0, 1]$
  - mean offer given F_t assumed to be higher for higher-type departments
- departments make only one offer to applicants
- $m = \sum_{j \in D} m_j$ graduates per department accept the highest offer received
  - $m_t = \sum_{j \in D_t} m_j$

Recap: Model

- symmetric strategy rule $\tilde{\pi}(x) = (\tilde{\pi}_1(x), \ldots, \tilde{\pi}_k(x))$
  - $\tilde{\pi}_t(x)$: probability of selecting tier t to choose an applicant
  - $\sum_{t=1}^{k} \tilde{\pi}_t(x) = 1, \forall x$
- probability of choosing a particular applicant after tier t chosen: $\frac{1}{m_t}$
- applicant payoff: offer x if offered, zero otherwise
- department payoff: value v_t of applicant's tier t if offer accepted, zero otherwise

Recap: Theorem

- series of constants $\{\pi_j\}_{j \in 1..k}$
- series of cutoffs $\{x_0^* = 1, x_1^*, x_2^*, \ldots, x_k^*\}$
- $\tilde{\pi}_j(x) = \begin{cases} \pi_j \prod_{j<i\leq t}(1-\pi_i) & j \leq t \\ 0 & \text{otherwise} \end{cases}$ for $x \in [x_t^*, x_{t+1}^*)$
- $\pi_t = \frac{\pi_{t-1}}{\pi_{t-1} + (\frac{v_t}{v_{t-1}})^{\frac{1}{n-1}} \frac{m_{t-1}}{m_t}}$
- $F(x_t) = F(x_{t-1}) - \frac{m_t}{\pi_t}(1 - (\frac{v_{t+1}}{v_t})^{\frac{1}{n-1}})$
  - $F(x) = \frac{1}{n_1}F_1(x) + \frac{1}{n_2}F_2(x) + \ldots$

Recap: Expressions

$$h_i^t = \sum_{s=t}^{k} \left[\left[\pi_t \prod_{j=t+1}^{s} (1-\pi_j) \right] \int_{x_s}^{x_{s-1}} Q_t(\tilde{x}) dF_i(\tilde{x}) \right] \tag{1}$$

- probability that a tier i department successfully hires a tier t graduate: P(department chooses them and they accept) for all offer intervals
- probability of applicant of tier t accepting an offer:
  $Q_t(x) = (1 - (F(x_{s-1}) - F(x))\frac{\pi_t}{m_t} \prod_{i=t+1}^{s} (1-\pi_i) - \sum_{k=t}^{s-1} (F(x_{k-1}) - F(x_k)\frac{\pi_t}{m_t} \prod_{i=t+1}^{k} (1-\pi_i)))^{n-1}$
- probability that department doesn't hire anyone: $1 - \sum_t h_i^t$

Estimation: Knowns and Unknowns

Known (under population dataset):

- $n = n_1 + \cdots + n_k + n_s$ hiring organizations per tier
- $m = m_1 + \cdots + m_k$ applicants per tier
- o_{ij} observed number of tier j applicants hired anywhere in tier i organizations

Unknown:

- $v = (v_1, \ldots, v_k)$ value of applicants from each tier
- $\mu = (\mu_1, \ldots, \mu_k, \mu_s)$ and $\sigma = (\sigma_1, \ldots, \sigma_k, \sigma_s)$ shape parameters defining $F_1(x), \ldots, F_k(x), F_s(x)$

Estimation: Mechanics

Determine the most likely v, μ and σ parameter vectors by minimizing the difference between theoretical and expected placements:

$$\min_{v,\mu,\sigma} \sum_{i,j} \frac{(o_{ij} - n_i h_{ij}(v,\mu,\sigma))^2}{n_i h_{ij}(v,\mu,\sigma)} \tag{2}$$

Population data is unknown so estimate using sample data.

Estimation: Type Allocation

Network is a directed multigraph (Peixoto, 2012)

- placement matrix $A : |D \cup S| \times |D \cup S|$
- $A_{i,j}$: number of applicants from dept i hired at dept j
- special property of sinks s: $\forall j \in [1, |D \cup S|], A_{s,j} = 0$

Define $\hat{A} = A[1 : |D \cup S|; 1 : |D|]$ as the truncated placement matrix

- sinks are incapable of graduating applicants

$\hat{A}_{i,j} \sim \text{Pois}(\lambda_{t_i,t_j})$ (Karrer and Newman, 2010)

- $t_i \in [1, k+1]$: tier of department i
- total of $k(k+1)$ Poisson means

Estimation: Type Allocation

Denote the likelihood (Peixoto, 2012) of $\hat{A}$ being produced by the $k(k+1)$ Poisson means as

$$\mathcal{L}(\lambda_{t_i,t_j,\forall i,j}) = \prod_{i=1}^{M}\prod_{j=1}^{N} \Pr(\text{Pois}(\lambda_{t_i,t_j}) = \hat{A}_{i,j}) \tag{3}$$

Rescale via log-likelihood as

$$\log(\mathcal{L}(\lambda_{t_i,t_j,\forall i,j})) = \sum_{i=1}^{M}\sum_{j=1}^{N} \log(\Pr(\text{Pois}(\lambda_{t_i,t_j}) = \hat{A}_{i,j})) \tag{4}$$

Estimated Poisson means solve $\max\limits_{\lambda_{t_i,t_j},\forall i,j} \log(\mathcal{L}(\lambda_{t_i,t_j,\forall i,j}))$

Estimation: Type Allocation

Maximum likelihood estimation adapted from Peixoto (2014):

- let $t_i \in [1, k] = 1, \forall i \in [1, |D|]$
- let $t_j = k + 1, \forall j \in [|D| + 1, |D \cup S|]$ be fixed (sinks)
- compute the current Poisson parameter estimate for $t_1 \in [1, k], t_2 \in [1, k + 1]$ as

$$\lambda_{t_1,t_2} = \frac{\sum_{i=1,t_i=t_1}^{|D|} \sum_{j=1,t_j=t_2}^{|D \cup S|} \hat{A}_{i,j}}{|\{i : t_i = t_1\}| * |\{j : t_j = t_2\}|} \tag{5}$$

- this is the mean number of placements observed from a t_1 department to a t_2 department

Estimation: Type Allocation

Markov Chain Monte Carlo stochastic gradient descent (Peixoto, 2014)

- compute the likelihood $\log(\mathcal{L}(\lambda_{t_i,t_j,\forall i,j}))$ using the current parameter estimate
- randomly, uniformly, choose an arbitrary department and assign it a type different from its current type
- compute the new estimated parameters and the new likelihood
- if the likelihood is greater than the previous likelihood, repeat the above; if not, seek a new random reassignment and then repeat the above

Method is a consistent estimator of true parameters (Bickel et al., 2013).

Estimation: Type Allocation

Finally, decide on the number of types k

- Minimum description length (Peixoto, 2013)
- trade-off greater likelihood of more parameters with decreasing likelihood of a naturally-generated complex network
- Analogous to Bayesian Information Criterion penalty term: $n \log(k)$

$$\mathcal{D} = \frac{k(k+1)}{2} log(\sum_{i,j} \hat{A}_{i,j}) + |D| log(k) \tag{6}$$

which modifies the problem to maximize

$$\mathcal{L}' = log(\mathcal{L}(\lambda_{t_i,t_j,\forall i,j}))) - \mathcal{D} \tag{7}$$

Estimation: Deep Parameters (Proof of Concept)

Recall: we must solve:

$$\min_{v,\mu,\sigma} \sum_{i,j} \frac{(o_{ij} - n_i h_{ij}(v,\mu,\sigma))^2}{n_i h_{ij}(v,\mu,\sigma)} \tag{8}$$

- o_{ij}: given by the type allocation
- n_i and m_i: given by true data

Expression is only valid for the true n_i and m_i if o_{ij} is at the population level, which it is not. Instead try to find the most likely sample-level n_i and m_i by making them part of the estimation:

$$\min_{v,\mu,\sigma,\vec{n},\vec{m}} \sum_{i,j} \frac{(o_{ij} - \vec{n}_i h_{ij}(v,\mu,\sigma,\vec{n},\vec{m}))^2}{\vec{n}_i h_{ij}(v,\mu,\sigma,\vec{n},\vec{m})} \tag{9}$$

subject to $\sum_i \vec{n}_i = n$, $\sum_i \vec{m}_i = m$ where n and m are exogenous.

Estimation: Deep Parameters

First-order condition does not have a closed form because $F(x)$ is a composition of truncated normal distributions and thus x_i^* has no closed form.

Use a numerical optimization method instead:

1. Fix n and m e.g. $n = 1550$ and $m = 1700$
2. Specify search ranges for each of the 22 parameters
3. Find the optimal parameter values along the ranges

- Use a numerical optimizer within the above optimizer to solve for x_i^* at each point

Estimation: Deep Parameters

```
pi_1 = 1.0
pi_2 = 0.44486241377030267
pi_3 = 0.23869896952551992
pi_4 = 0.3462336238398713

x_0 = 1.0 (error: 0.0)
x_1 = 0.7078283281268729 (error: 3.884104031417403e-19)
x_2 = 0.4514348894411525 (error: 6.595429340822289e-22)
x_3 = 0.2163706881255148 (error: 2.1899043600865277e-21)
x_4 = 0.0 (error: 0.0)

F(x_0) = 1.0
F(x_1) = 0.9554867223256124
F(x_2) = 0.6719797556994577
F(x_3) = 0.09343724417465771

objective value = 8.361717436501587

estimated n_i = [90.0, 156.0, 350.0, 80.0, 873.0]
estimated m_i = [470.0, 377.0, 265.0, 588.0]

estimated Q_t(x_s) (rows are offer value from high to low, columns are type in increasing type index:
[1.0, 1.0, 1.0, 1.0]
[0.8635, 1.0, 1.0, 1.0]
[0.5139, 0.5951, 1.0, 1.0]
[0.2294, 0.2657, 0.4464, 1.0]

estimated h_it:
[0.6163, 0.1464, 0.0277, 0.0165]
[0.3706, 0.2253, 0.0597, 0.0211]
[0.2153, 0.1986, 0.1133, 0.0128]
[0.1397, 0.1279, 0.1036, 0.1107]
[0.1959, 0.1793, 0.1108, 0.0342]
```

Estimation: Deep Parameters

v_1: 1, v_2: 0.86, v_3: 0.51, v_4: 0.23