Computational Methods in Economics

Algorithms

January 14, 2020

Computing and Algorithms

- Algorithm: well-defined computational procedure that takes some value as **input** and produces some value as **output**.
- Designing an algorithm is about finding the most **efficient** way to create an input/output relationship.

Stable Matching (1)

- Gale and Shapley (1962): Can we design a college admissions process or a job recruiting process that was self-enforcing?
- Consider the marriage problem.
- A set of men and women desire to form partnerships.
- Each woman has a preference ordering over men.
- Each man has a preference ordering over women.
- Based on those preferences, women make offers to men, and marriages happen.

Stable Matching (2)

Initially all m in M and w in W are free.

```
While there is a man m who is free and hasn't
proposed to every woman w
Choose such a man m
Let w be the highest-ranked woman in m's preference list
to which m has not yet proposed
   If w is free then
   (m, w) become engaged
   Else w is currently engaged to m'
   If w prefers m' to m then
     m remains free
   Else w prefers m to m'
     (m, w) become engaged
     m' becomes free

Endif
   Endif
Endwhile
```

Constraints

- Physical constraints (RAM, storage).
- Worst-case and average-case analysis. (Benchmark)
- Order of growth (scale).

Structure

- Initialization.
- Maintenance.
- Termination.

Design

- Incremental approach
- Divide and Conquer

Divide-and-conquer approach

- **Divide** the problem into a number of subproblems that are smaller instances of the same problem.
- **Conquer** the subproblems by solving them recursively.
- **Combine** the solutions to the subproblems into the solution for the original problem.

Examples

- Search and Matching
- EM algorithm

Theory

Example 1

Example 2

Computation

Searching, Matching and Migrating: Space and population

- Geography
  - **Discrete** set $\mathcal{J}$ of J cities j where people both live and work
  - Exogenous sites summarized by the distance matrix $d = (d_{jl})$
  - Endogenous **population** size L_j and **number of firms** N_j
  - Some cities may disappear in equilibrium
- Land market
  - Discrete choice: **one unit** of land per individual
  - Competition between **commercial** and residential real estate
  - Firms, vacant or active, consume ψ units

$$R_j = R(L_j + \psi N_j)$$

- All individuals are **workers**
  - Fixed total population (normalized to 1)
  - Endowed with individual **ability** $h \sim \ell(\cdot) = U([0,1])$
  - Die at rate Ξ (perpetual youth, exit value set to zero)

Searching, Matching and Migrating: Workers

- Set-up
  - Maximize total discounted (at rate r) lifetime income
  - **Unemployed** $i = u$ are in measure $u_j(h)$ and earn b (fixed)
  - **Employed** $i = e$ are in measure $\ell_j(h) - u_j(h)$ and earn w
  - All workers are endowed with bargaining power β
  - Mobility cost $\phi_{jl} = \phi(d_{jl})$ between two cities

- Initial location problem
  - New urban workers, in measure $\omega_j(h)$, start off as unemployed
  - **Reshuffling** is irrespective of skills (no skill dynasties):

$$\omega_j(h) = \Xi \ell(h) L_j$$

- Assumption: workers may only migrate upon finding a job
  - Means that mobility costs (or housing market frictions) are very high for unemployed workers

Searching, Matching and Migrating: Firms

- Set-up
  - Distribution of latent **productivity** $p \sim g(\cdot) = U([0,1])$
  - Firms equate **jobs**: not a theory of firm size
  - Value of a type-p vacancy: $\Pi_j^v(p)$
  - If matched, **location-invariant** production technology $f(h,p)$
- Vacancies
  - Local measure of type-p vacancy $v_j(p)$
  - Total number of vacancies $V_j = \int v_j(p)dp$
  - **Selection**: lower bound $\underline{p}_j$ s.t. $\Pi_j^v(\underline{p}_j) = 0$
- Matches
  - Joint match distribution $m_j(h,p)$
  - Exogenous destruction at rate δ
  - Local accounting relationship $L_j - U_j \equiv N_j - V_j$

Searching, Matching and Migrating: Search

- Set-up
  - Continuous time, infinite horizon, **random** search
  - Both off-the-job and on-the-job & both within and between cities
- Jobseekers
  - Intensity-adjusted measure E_j of jobseekers in city j
  - $\varrho_{lj} = \varrho(d_{jl}) \in [0,1]$ **connectedness** of workers in l to jobs in j
  - ζ relative search efficiency of employed workers

$$E_j = U_j + \zeta(L_j - U_j) + \sum_{k \neq j} \varrho_{kj} \left[U_k + \zeta(L_k - U_k)\right]$$

- Matching
  - $\mathcal{M}(E_j, V_j)$ number of matches in j and $\theta_{lj} = \frac{\varrho_{lj}\mathcal{M}(E_j,V_j)}{E_j \times V_j}$

Meeting rates

	Worker $(i,j) \to$ Vacancy in l (independent of h)	Vacancy in $j \to$ Worker (i,l) (independent of p)
$i = u$	$\theta_{jl} v_l(p)$	$\theta_{lj} u_l(h)$
$i = e$	$\zeta\theta_{jl} v_l(p)$	$\zeta\theta_{lj} m_l(h,p')$

Searching, Matching and Migrating: Value functions and surplus

- Agents' value functions
  - The value for a type-p firm in city j when matched with a type-h worker paid at wage w is $\Pi_j^f(h, w, p)$
  - The corresponding value for the worker is $\mathcal{V}_j^e(h, w, p)$
  - The value of a type-h unemployed worker in city j is $\mathcal{V}_j^u(h)$
- Match product (**transferability**)

$$\forall w, \;\; \mathbb{P}_j(h, p) \equiv \Pi_j^f(h, w, p) + \mathcal{V}_j^e(h, w, p)$$

- Surplus

$$\mathcal{S}_{jl}^*(h, p) = \mathbb{P}_l(h, p) - \mathcal{V}_j^u(h) - \Pi_l^v(p)$$

Searching, Matching and Migrating: Wage determination

- Bargaining à-la Lise et al. (2016) over gross match surplus
  - But match occurrence depends on net surplus

- Reservation-wage strategy $\psi_{jl}(h,p)$:

$$\forall p \in \mathcal{P}^0_{jl}(h) \equiv \{y : \mathcal{S}_{jl}(h,y) \geq 0\},$$
$$\mathcal{V}^e_l(h,\psi_{jl}(h,p),p) - \mathcal{V}^u_j(h) = \beta \mathcal{S}^*_{jl}(h,p)$$
$$\forall h \in \mathcal{H}^0_{jl}(p) \equiv \{x : \mathcal{S}_{jl}(x,p) \geq 0\},$$
$$\Pi^f_l(h,\psi_{jl}(h,p),p) - \Pi^v_l(p) = (1-\beta)\mathcal{S}^*_{jl}(h,p)$$

- Firm-switching wage-change strategy $\psi_{jl}(h,p,p')$:

$$\forall p' \in \mathcal{P}^c_{jl}(h,p) \equiv \{y : \mathcal{S}_{jj}(h,p) \leq \mathcal{S}_{jl}(h,y)\},$$
$$\mathcal{V}^e_l(h,\psi_{jl}(h,p,p'),p') - \mathcal{V}^u_j(h) = \mathcal{S}^*_{jj}(h,p) + \beta[\mathcal{S}^*_{jl}(h,p') - \mathcal{S}^*_{jj}(h,p)]$$
$$\forall p' \in \mathcal{P}^f_{jl}(h,p) \equiv \{y : \mathcal{S}_{jj}(h,y) \leq \mathcal{S}_{jl}(h,p)\},$$
$$\Pi^f_l(h,\psi_{jl}(h,p',p),p) - \Pi^v_l(p) = (1-\beta)[\mathcal{S}^*_{jl}(h,p) - \mathcal{S}^*_{jj}(h,p')]$$

Searching, Matching and Migrating: Bellman equations

- Lifetime discounted **income** of a type-h unemployed worker in city j:

$$(r+\Xi)\mathcal{V}_j^u(h) = b - R_j + \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}_{jk}^0(h)} \left[\mathcal{V}_k^e(h,\psi_{jk}(h,y),y) - \mathcal{V}_j^u(h)\right] v_k(y)dy$$

- Expected discounted **profit** of a type-p vacancy in city j:

$$r\Pi_j^v(p) = -\psi R_j + \sum_{k\in\mathcal{J}} \theta_{kj} \int_{\mathcal{H}_{kj}^0(p)} \left[\Pi_j^f(x,\psi_{kj}(x,p),p) - -\Pi_j^v(p)\right] u_k(x)dx$$

$$+\zeta \sum_{k\in\mathcal{J}} \theta_{kj} \int_{\mathcal{H}_{kj}^0(p)} \int_{\mathcal{P}_{kj}^f(x,p)} \left[\Pi_j^f(x,\psi_{kj}(x,y,p),p) - (1-\beta)\phi_{jk} - \Pi_j^v(p)\right] m_k(x,y)dydx$$

- **Product** of a type-(h,p) match in city j:

$$r\mathbb{P}_j(h,p) = f(h,p) - (1+\psi)R_j - \delta\left[\mathbb{P}_j(h,p) - (\mathcal{V}_j^u(h) + \Pi_j^v(p))\right] - \Xi\left[\mathbb{P}_j(h,p) - \Pi_j^v(p)\right]$$

$$-\zeta \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}_{jk}^c(h,p)} \left[\mathbb{P}_j(h,p) - \left(\mathcal{V}_k^e(h,\psi_{jk}(h,p,y),y) + \Pi_j^v(p)\right)\right] v_k(y)dy$$

Searching, Matching and Migrating: Steady state

- Destruction rate for a type-(h,p) match in city j:

$$\delta_j(h,p) = \delta + \Xi + \zeta \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}^c_{jk}(h,p)} v_k(y)dy$$

- Motion laws on the labor market:

$$u_j(h)\left(\Xi + \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}^0_{jk}(h)} v_k(y)dy\right) = \delta \int m_j(h,y)dy + \omega_j(h)$$

$$m_j(h,p)\delta_j(h,p) = v_j(p) \sum_{k\in\mathcal{J}} \theta_{kj} \left(1_{h\in\mathcal{H}^0_{kj}(p)} u_k(h) + \zeta \int_{\mathcal{P}^f_{kj}(h,p)} m_k(h,y)dy\right)$$

$$v_j(p) \sum_{k\in\mathcal{J}} \theta_{kj} \int_{\mathcal{H}^0_{kj}(p)} \left(u_k(x) + \zeta \int_{\mathcal{P}^f_{kj}(x,p)} m_k(x,y)dy\right) dx = \int \delta_j(x,p) m_j(x,p) dx$$

Searching, Matching and Migrating: "Closed" forms

- Conditional on surplus, closed forms for unmatched agents:

$$
\begin{aligned}
(r+\Xi)\mathcal{V}_j^u(h) &= b - R_j + \beta \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}^0_{jk}(h)} \mathcal{S}_{jk}(h,y) v_k(y) dy \\
r\Pi_j^v(p) &= -\psi R_j + (1-\beta) \sum_{k\in\mathcal{J}} \theta_{kj} \int_{\mathcal{H}^0_{kj}(p)} \mathcal{S}_{kj}(x,p) u_k(x) dx \\
&+ (1-\beta)\zeta \sum_{k\in\mathcal{J}} \theta_{kj} \int_{\mathcal{H}^0_{kj}(p)} \int_{\mathcal{P}^f_{kj}(x,p)} \left[\mathcal{S}_{kj}(x,p) - \mathcal{S}_{kk}(x,y)\right] m_k(x,y)
\end{aligned}
$$

- This allows us to recover a closed-form expression for match product:

$$
\begin{aligned}
r\mathbb{P}_j(h,p) &= f(h,p) - (1+\psi)R_j - (\delta+\Xi)\mathcal{S}_{jj}(h,p) - \Xi\mathcal{V}_j^u(h) \\
&+ \zeta \sum_{k\in\mathcal{J}} \theta_{jk} \int_{\mathcal{P}^c_{jk}(h,p)} \beta \left[\mathcal{S}_{jk}(h,y) - \mathcal{S}_{jj}(h,p)\right] v_k(y) dy
\end{aligned}
$$

Searching, Matching and Migrating: Operationalization

Model	Parameterization
R_j	$R_0 + R_1(L_j + \psi N_j)^{R_2}$
$f(h, p)$	$[h^{(\rho-1)/\rho} + p^{(\rho-1)/\rho}]^{\rho/(\rho-1)}$
ϱ_{jl}	$\exp(-\varrho d_{jl})$
$\mathcal{M}(E_j, V_j)$	$m_0 E_j^{m_1} V_j^{1-m_1}$
ϕ_{jl}	$1_{j \neq l} \times \phi$

Algorithm

For a given $\Theta = \big(\underbrace{\Xi, r}_{\text{Life}}, \underbrace{\psi, R_0, R_1, R_2}_{\text{Land}}, \underbrace{b, \beta, \delta, \zeta, \rho, m_0, m_1}_{\text{Labor}}, \underbrace{\varrho, \phi}_{\text{Space}}\big)$,

1. Assume $\mathcal{S}_{jl}(h,p)$, $u_j(h)$, $m_j(h,p)$ and θ_{jl} known
2. Compute $\ell_j(h)$, L_j, U_j, V_j (from θ_{jl}), N_j and R_j
3. Construct $\Pi_j^v(p)$
4. Find $\underline{p}_j$ from the zero-profit cutoff condition
5. Recover $v_j(p)$
6. Update $u_j(h)$ and $m_j(h,p)$ using steady state conditions
7. Construct $\mathcal{V}_j^u(h)$ and $\mathbb{P}_j(h,p)$ and update $\mathcal{S}_{jl}(h,p)$ from its definition
8. Repeat steps 3 to 7 until convergence.

Theory

Example 1

Example 2

Computation

Discrete Mixing Distributions (1)

- Consider a discrete choice model
- An agent i obtains utility U_{ijt} from product j at time t.

$$U_{ijt} = \beta_i x_{ijt} + \epsilon_{ijt}$$

  with
  - ϵ_{ijt} is iid and extreme value distributed.
  - $\beta_i \sim f(\beta \mid \theta)$ where θ represents the parameters of the distribution
- Let y_{it} denote the alternative that agent i chooses in situation t.

Mixing Distributions (2)

- Conditional on β, the probability of y_i is

$$K_i(\beta) = \prod_t \frac{\exp(x_{iy_it})}{\sum_j \exp(\beta x_{ijt})} \tag{1}$$

- The choice probability is

$$P_i(\theta) = \int K_i(\beta) f(\beta \mid \theta) d\beta$$

Mixing Distributions (3)

- For discrete distribution with support at β_c, $c = 1, \ldots, C$, the choice probability is

$$P_i(\theta) = \sum_c s_c K_i(\beta_c) \tag{2}$$

where $s_c = f(\beta_c \mid \theta)$ is the share of the population that has coefficients β_c.

Mixing Distributions (4): EM

- Select initial values β_c^0 and s_c^0 $\forall$ c. Probably start with equal shares and estimates from a logit for partitions of the sample into C subsamples.
- Calculate the weights as

$$h_{ic}^0(\theta^0) = \frac{s_c^0 K_n(\beta_c^0)}{\sum_m s_m^0 K_n(\beta_m^0)} \tag{3}$$

- Update the shares as

$$s_c^1 = \frac{\sum_n h_{ic}^0}{\sum m \sum_i h^0} \tag{4}$$

- Run C standard logits on the data using weights h_{nc}^0 in the c_{th} run, and derive β_c^1

Practice

- Number of iterations.
- Size of the update step.
- Convergence criteria.

Theory

Example 1

Example 2

Computation

Fundamentals

- ?solve or help(solve).
- R is case sensitive.
- “¡-” is an assignment operator.
- Avoid loops in R.