A Study on Optimal Assignment problems

Doubly stochastic matrices

Satwik (193079009)

Department of Electrical Engineering
Indian Institute of Technology Bombay

11^{th} May, 2021.

What is an assignment problem?

- An integer program problem
- N individuals and N jobs and a time taken by the N individuals for the N jobs are known, how can we assign the jobs such that total time for the N jobs is minimized and one job is assigned to only one individual?
- $$\begin{array}{l} \\ \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Job 1} & \text{Job 2} & \text{Job 3} \end{array} \\ \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \end{array}$$
- Find all the possible arrangements $(t_{11}, t_{22}, t_{23}), (t_{12}, t_{21}, t_{23}), \ldots$ etc. and find the total time for each of the arrangements and thereby find the solution

What is an assignment problem?

- The number of such arrangements = number of possible 3×3 permutation matrices = $3! = 6$
- If an organization has 20 individuals and 20 jobs $\Rightarrow 20! = 2.43 \times 10^{18}$ arrangements are possible
- An organization has 300 individuals and 30 different jobs
- These are some examples of personnel-assignment problems

Matchings

- Assignment problems can also be expressed in graph notation
- A square matrix is said to be doubly stochastic if its elements are non-negative and all row sums and column sums are equal one.
- A permutation matrix is an example of a doubly stochastic matrix
- Each of the arrangements $(t_{11}, t_{22}, t_{23}), (t_{12}, t_{21}, t_{23}), \ldots$ can be represented using a permutation matrix
- A permutation matrix represents a perfect matching between the sets $V = \{$Individual 1, Individual 2, Individual 3$\}$ and $W = \{$Job 1, Job 2, Job 3$\}$ and has an associated total cost (or time)
- $$\begin{array}{l} \\ \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Job 1} & \text{Job 2} & \text{Job 3} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \Rightarrow \mathit{Totaltime} = \mathrm{t}_{11} + t_{22} + t_{33}$$

Matchings cont.

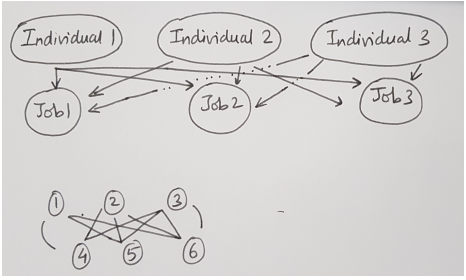
- $$\begin{array}{c} \\ \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Job 1} & \text{Job 2} & \text{Job 3} \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$ also represents a matching but not a perfect matching

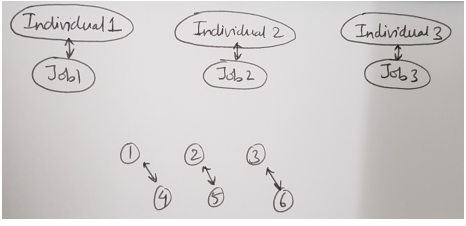
- We can consider the above problem as a bipartite graph G between V and W i.e. $G = (V, W; E)$ where E are the edges in the G
- The assignment problem can be written as

$$min \left\{ \sum_{(v,w) \in M} c(v, w) : \text{M is a perfect matching} \right\}$$

  where $c(v, w)$ represents the cost related to the edge $e = (v, w)$ for $v \in V$ and $w \in W$.
- When every vertex in G is connected to exactly one edge in M, then M is a perfect matching

Matching cont.





Konig, Egervary, and Kuhn

- In 1953, Harold Kuhn encountered Konig's work on matching problem on bipartite graphs and Egervary's analysis on covering systems.
- Using the results from the works, Kuhn in 1955 had proposed an algorithm for solving such optimization problems. He named the method as the 'the hungarian method' after the two Hungarian mathematicians Denes Konig and Jeno Egervary

Theorem 1

Theorem (Konig's Theorem 1)

In a bipartite graph G, the size of a maximum matching of G is equal to the size of a minimum covering of G.

$$max|M| = min|C|$$

- A covering is defined to be the set of straight lines necessary in order to cover all the matched entries in a matrix.
- The theorem suggests a dual between the number of covering lines of a matrix and the maximum matching in a bipartite graph

Example

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$ has 2 matched vertices and also requires 2 covering lines

Theorem 2

Theorem (Konig's Theorem 2)

Let $G = (V, W ; E)$ be a bipartite graph. G contains a perfect matching (marriage) if and only if $|V| = |W|$ and for all subsets V^* of V

$$|V^*| \leq |N(V^*)|$$

- Where $N(v)$ denotes the set of neighbours of a vertex $v \in V$, i.e. the set of vertices $w \in W$ that are connected to v by an edge $e \in E$
- This theorem tells us the condition for existence of a perfect matching. For vertices in V there should be at least the same amount of vertices available in W for a perfect matching

Theorem 3

Theorem (Egervary Theorem 1)

If (a_{ij}) is an $n \times n$ matrix of non-negative integers then, subject to condition $\lambda_i + \mu_j \geq a_{ij}$ $(i, j = 1, 2, \ldots, n)$, we have

$$min \sum_{k=1}^{n} (\lambda_k + \mu_k) = max \sum_{i=1}^{n} a_{ip(i)}$$

- where p is a permutation of the integers $\{1, 2, \ldots, n\}$
- A dual relation: minimizing the covering will result in maximization of matching

Hungarian algorithm

1. Identify the smallest element of each row ($u_i = \min\ c_{ij}$ for $1 \leq j \leq n$) and subtract it from all elements of that row
2. Identify the smallest element of each column ($v_j = \min\ c_{ij}$ for $1 \leq i \leq n$) and subtract it from all elements of that column it from all elements of that column
3. Initialize with any (λ_i) and (μ_j) satisfying $\lambda_i + \mu_j \geq c_{ij}$ $(i, j = 1, 2, \ldots, n)$
4. Find a maximum matching M (Theorem 1) in the subgraph $G(\lambda, \mu)$ of G that only contains the edges of E that satisfy $\lambda_i + \mu_j \geq c_{ij}$
5. If M is perfect matching then stop
6. Else $G(\lambda, \mu)$ must contain (Theorem 2) a subset $|V^{'}| \subseteq V$ such that $|V^{'}| > |V(U^{'})|$: update the covering system through (Theorem 3) and go to step 4

$$\begin{cases} \lambda_i = \lambda_i - \min\ \{\lambda_i + \mu_j - a_{ij}\} & : i \in V^{'}, j \in N(V^{'}) \\ \mu_j = \mu_j - \min\ \{\lambda_i + \mu_j - a_{ij}\} & : i \in V^{'}, j \in N(V^{'}) \end{cases}$$

Example

$$C = \begin{pmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{pmatrix} \tag{1}$$

$$u = \begin{pmatrix} 75 \\ 35 \\ 90 \\ 45 \end{pmatrix} \Rightarrow C^* = \begin{pmatrix} 15 & 0 & 0 & 5 \\ 0 & 50 & 20 & 30 \\ 35 & 5 & 0 & 15 \\ 0 & 65 & 50 & 70 \end{pmatrix} \tag{2}$$

$$v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \end{pmatrix} \Rightarrow C^{**} = \begin{pmatrix} 15 & 0 & 0 & 0 \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{pmatrix} \tag{3}$$

Example 2 contd.

$$\begin{pmatrix} 15 & 0^* & 0 & 0 \\ 0^* & 50 & 20 & 25 \\ 35 & 5 & 0^* & 10 \\ 0 & 65 & 50 & 65 \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} 35 & 0 & 0 & 0 \\ 0 & 30 & 0 & 5 \\ 55 & 5 & 0 & 10 \\ 0 & 45 & 30 & 45 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} 35 & 0^* & 0 & 0 \\ 0 & 30 & 0 & 5 \\ 55 & 5 & 0^* & 10 \\ 0^* & 45 & 30 & 45 \end{pmatrix} \tag{6}$$

Example 2 contd.

$$\begin{pmatrix} 40 & 0 & 5 & 0 \\ 0 & 25 & 0 & 0 \\ 55 & 0 & 0 & 5 \\ 0 & 40 & 30 & 40 \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} 40 & 0^* & 5 & 0 \\ 0 & 25 & 0 & 0^* \\ 55 & 0 & 0^* & 5 \\ 0^* & 40 & 30 & 40 \end{pmatrix} \tag{8}$$

Open shop scheduling

- Same as the personnel assignment problem (here we assign jobs to machines)
- With two key changes:
- We have a non-negative integer matrix T gives the total amount of time, t_{ij} , job j must be processed on machine i $(i, j = 1, 2, \ldots, n)$
- Each processing can be interrupted at any time and resumed later

Algorithm

1. Identify the maximum of row and column sums of all the rows and columns (c^*)
2. for i = 1 to n and for j = 1 to n

$$s_{ij} = \min\,(a_i, b_j)$$
$$a_i = a_i - s_{ij}$$
$$b_j = b_j - s_{ij}$$

   where $a_i = c^* - i^{th}$ row sum and $b_j = c^* - j^{th}$ column sum
3. Modify $T = T + S$
4. let $G = (U, V; E)$ be a bipartite graph with $|U| = |V|$ and $E = [i, j] : t_{ij} > 0$
5. Find a perfect matching in G, and the corresponding permutation matrix P, here coefficient = the minimum value in T corresponding to the permutation matrix
6. $\tau = \min\{t_{ij} : p_{ij} = 1\} \Rightarrow T = T - \tau P$
7. if T is a zero matrix then stop else go to step 4

Algorithm cont.

```python
import numpy as np
import itertools
from networkx import from_numpy_matrix
from networkx.algorithms.bipartite.matching import maximum_matching

if __name__=='__main__':
    D = np.matrix([[0.1,0.2,0.7],[0.3,0.4,0.3],[0.6,0.4,0]])
    print(D)
    print()
    D = D.astype('float')
    birk_coeff = []
    birk_permutations = []
    m, n = D.shape
    if m != n:
        raise ValueError('given matrix is not a square matrix'.format(m, n))
    indices = list(itertools.product(range(m), range(n)))
    while not np.all(D == 0):
        replica = np.zeros_like(D)
        replica[D.nonzero()] = 1

        bipartite = np.vstack((np.hstack((np.zeros((m, m)), replica)), np.hstack((replica.T, np.zeros((n, n))))))
        bipartite_graph = from_numpy_matrix(bipartite)
        left_nodes = range(n)

        perfect_matching_graph = maximum_matching(bipartite_graph, left_nodes)
        perfect_matching_nodes = {u: v % n for u, v in perfect_matching_graph.items() if u < n}

        num = len(perfect_matching_nodes)
        P = np.zeros((num, num))
        P[list(zip(*(perfect_matching_nodes.items())))] = 1
        c = min(D[i, j] for (i, j) in indices if P[i, j] == 1)
        birk_coeff.append(c)
        birk_permutations.append(P)
        D -= c * P
        D[np.abs(D) < np.finfo(np.float).eps * 10.] = 0.0

    answer = list(zip(birk_coeff, birk_permutations))
    for i in answer:
        print('coeff:',np.around(i[0],2))
        print(i[1])
```

Algorithm cont.

```
[[0.1 0.2 0.7]
 [0.3 0.4 0.3]
 [0.6 0.4 0. ]]

coeff: 0.4
[[0. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]]
coeff: 0.1
[[1. 0. 0.]
 [0. 0. 1.]
 [0. 1. 0.]]
coeff: 0.2
[[0. 1. 0.]
 [0. 0. 1.]
 [1. 0. 0.]]
coeff: 0.3
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]]
```

Example

	1	2	3	4	5	6	7
tuneup	25	–	20	–	–	–	–
oil change	15	15	–	25	–	–	–
rotate tires	25	15	10	15	–	–	–
dummy job 1	–	–	–	–	–	–	–
dummy job 2	–	–	–	–	–	–	–
dummy job 3	–	–	–	–	–	–	–
dummy job 4	–	–	–	–	–	–	–

Example

$$\begin{bmatrix} 25 & 0 & 20 & 0 & 20 & 0 & 0 \\ 15 & 15 & 0 & 25 & 0 & 10 & 0 \\ 25 & 15 & 10 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 35 & 20 & 0 \\ 0 & 35 & 0 & 0 & 10 & 20 & 0 \\ 0 & 0 & 35 & 0 & 0 & 15 & 15 \\ 0 & 0 & 0 & 15 & 0 & 0 & 50 \end{bmatrix}$$

Example

```
[[25  0 20  0 20  0  0]
 [15 15  0 25  0 10  0]
 [25 15 10 15  0  0  0]
 [ 0  0  0 10 35 20  0]
 [ 0 35  0  0 10 20  0]
 [ 0  0 35  0  0 15 15]
 [ 0  0  0 15  0  0 50]]

coeff: 10.0
[[1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 1.]]
coeff: 5.0
[[1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1.]]
coeff: 10.0
[[1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0. 0. 0.]
```

Example

It may be verified that the above matrix can be expressed as

$$15B_1 + 10B_2 + 10B_3 + 10B_4 + 5B_5 + 5B_6 + 5B_7 + 5B_8,$$

where B_i, $i = 1, 2, \ldots, 8$ are permutation matrices corresponding to the following permutations, respectively : (5416237), (1234567), (1645237), (3125674), (3215674), (1425637), (3415267), (5146237). The following summarizes the corresponding scheduling operations:

	15 m	10 m	10 m	10 m	5 m	5 m	5 m	5 m
tuneup	–	C_1	C_1	C_3	C_3	C_1	C_3	–
oil change	C_4	C_2	–	C_1	C_2	C_4	C_4	C_1
rotate tires	C_1	C_3	C_4	C_2	C_1	C_2	C_1	C_4

References


R.B.Bapat, T.E.S. Raghavan (1997)
Nonnegative matrices and applications
Cambridge university press


Mehlum, Maria. (2012)
Doubly stochastic matrices and the assignment problem.
MS thesis


Martello, Silvano. (2010)
Jenő Egerváry: from the origins of the Hungarian algorithm to satellite communication.
Central European Journal of Operations Research 18.1


Robert J. Vanderbei. (2020)
Linear Programming: Foundations and Extensions.
Springer Natural


Parthasarathy, T., and Sujatha Babu. (2012)
Stochastic games and related concepts.
Hindustan Book Agency

Thank You