Use of Pandas for data-analysis in allocation problems

Saurabh Kumar, Shana Moothedath & Madhu N. Belur,

Department of Electrical Engineering,

Indian Institute of Technology Bombay

SciPy, 15-Dec-15

Outline

1. Two typical allotment problems
2. Two separate groups to be allotted
   - Course to TA (Teaching Assistant)
   - Exam-centre to invigilator
3. Allotment versus data-analysis
4. Pandas

Allocation/Matching Problems

Given two sets U and V: find a matching between U and V.

Perfect matching: each node is matched

(requires U and V to be same size)

Often: many perfect matchings: find 'best' in some sense

Allocation/Matching Problems

Given two sets U and V: find a matching between U and V.

Perfect matching: each node is matched

(requires U and V to be same size)

Often: many perfect matchings: find 'best' in some sense

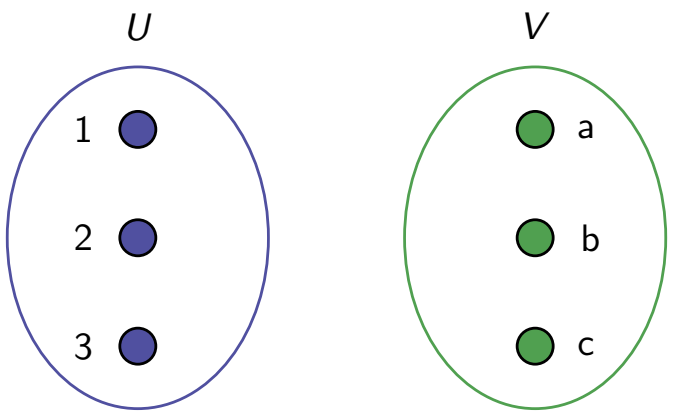
Allocation/Matching Problems

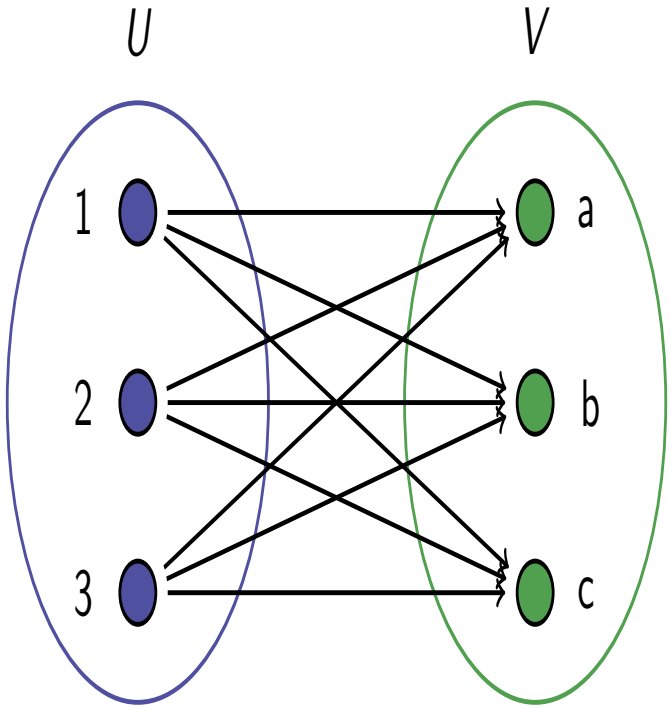
Given two sets U and V: find a matching between U and V.

Perfect matching: each node is matched

(requires U and V to be same size)

Often: many perfect matchings: find 'best' in some sense



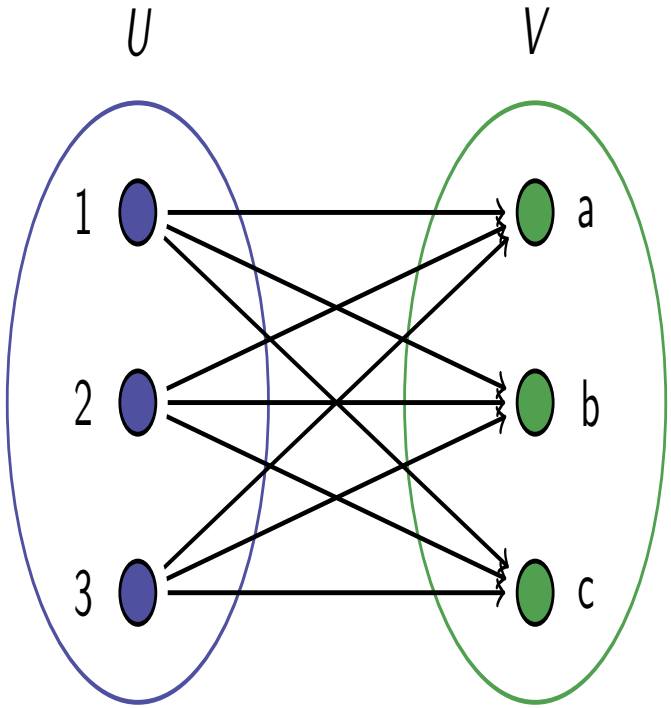


U and V have no common node. Each edge: one node in U and one in V

'Bipartite graph'

Other examples: Hospital-patient allocation, student-college admission

But not: room-partner allotment in hostels (non-bipartite, more-complex)

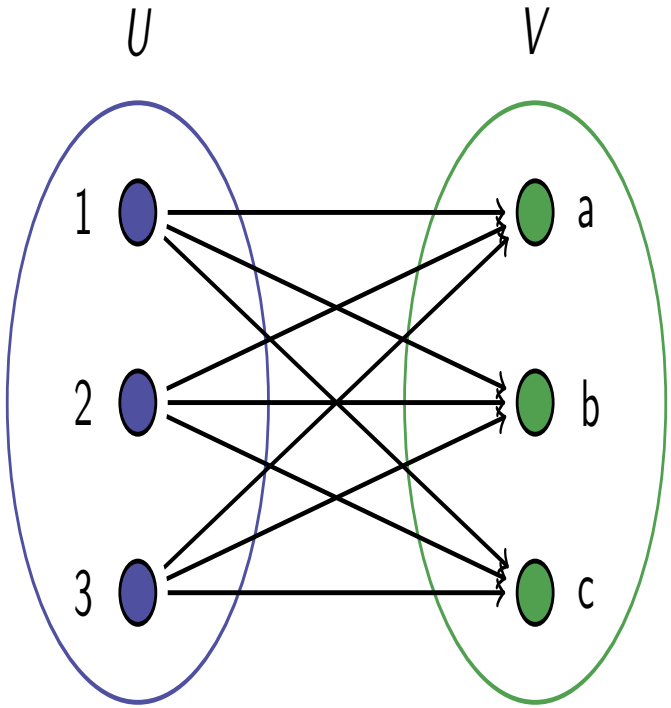


U and V have no common node. Each edge: one node in U and one in V

'Bipartite graph'

Other examples: Hospital-patient allocation, student-college admission

But not: room-partner allotment in hostels (non-bipartite, more-complex)

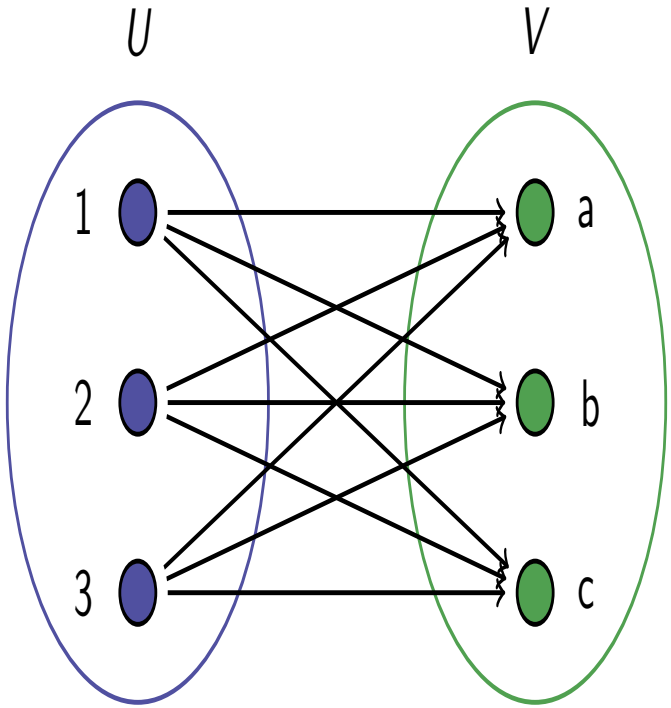


U and V have no common node. Each edge: one node in U and one in V

'Bipartite graph'

Other examples: Hospital-patient allocation, student-college admission

But not: room-partner allotment in hostels (non-bipartite, more-complex)

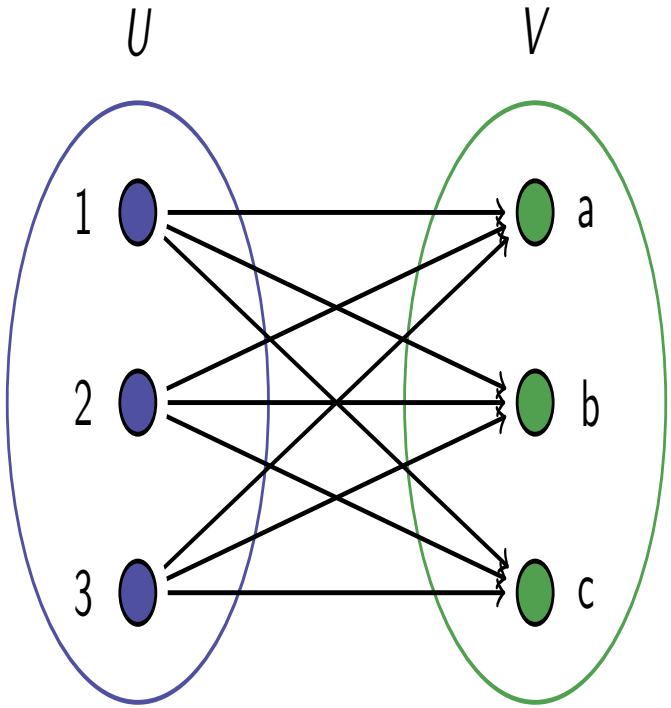


U and V have no common node. Each edge: one node in U and one in V

'Bipartite graph'

Other examples: Hospital-patient allocation, student-college admission

But not: room-partner allotment in hostels (non-bipartite, more-complex)

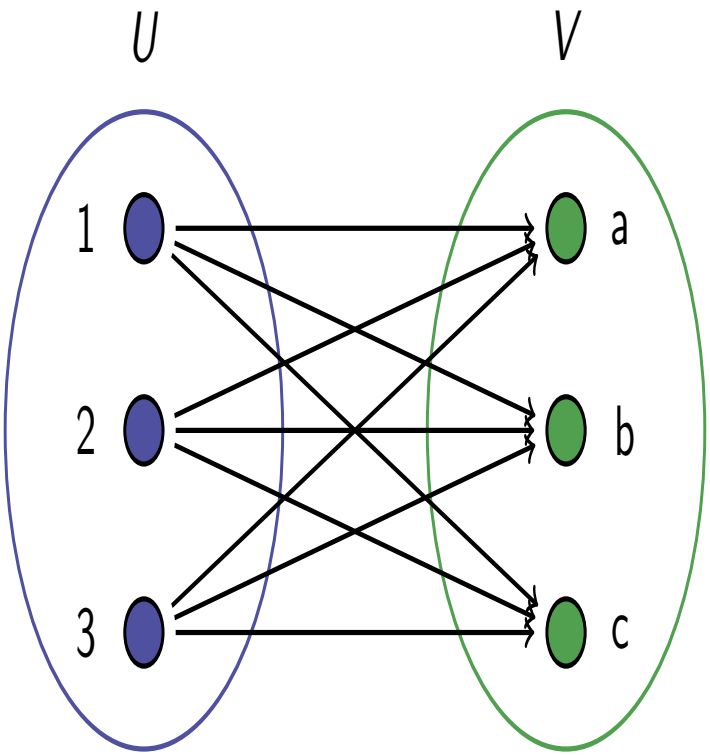


U and V have no common node. Each edge: one node in U and one in V

'Bipartite graph'

Other examples: Hospital-patient allocation, student-college admission

But not: room-partner allotment in hostels (non-bipartite, more-complex)

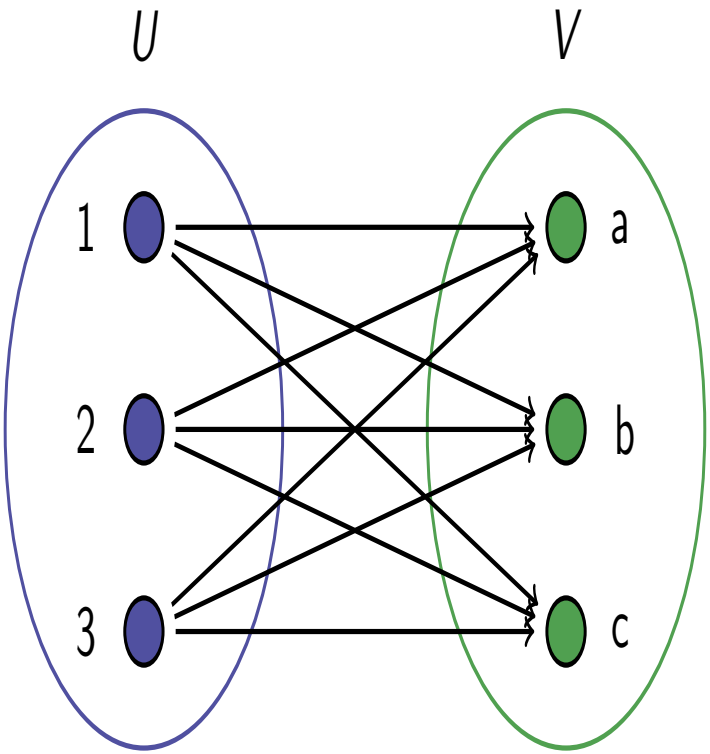


$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 5 & 10 & 15 \\ 10 & 5 & 35 \\ 15 & 40 & 10 \end{bmatrix} \end{array}$$

Edges have weights: preferences, eligibilities

Find maximum 'total weight match':

Allotter's decision versus fairness



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \underline{5} & 10 & 15 \\ 10 & 5 & \underline{35} \\ 15 & \underline{40} & 10 \end{bmatrix} \end{array}$$

Maximum total for 1-a, 2-c and 3-b, even though 1-a value is least

1st Allocation Problem: TA (Teaching Assistant) to course

Given two sets: students S and courses C: allot S and C

1. Data: students: course preferences, their score
2. Courses: number of slots: number of TA positions

   Eligibility (student performance/specialization)

3 parties: students, course instructors, HoD (administrator)

Student: each wants 1st choice

Course instructor: wants high score student as TA

Administrator ensure: all students/slots are allotted

students/courses are 'satisfied' most

1st Allocation Problem: TA (Teaching Assistant) to course

Given two sets: students S and courses C: allot S and C

1. Data: students: course preferences, their score
2. Courses: number of slots: number of TA positions

   Eligibility (student performance/specialization)

3 parties: students, course instructors, HoD (administrator)

Student: each wants 1st choice

Course instructor: wants high score student as TA

Administrator ensure: all students/slots are allotted

students/courses are 'satisfied' most

2nd allocation Problem: invigilator/centre

GATE: national level competitive exam: critical for IITs

Two sets: invigilators $\mathbf{I}$ and examination centres C: allot $\mathbf{I}$ to C

1. Data: invigilators give preference cities.

   Some invigilators are more experienced/senior.

2. Centre: number of invigilators required

   Centres have different experience

Three parties: invigilators, centres, Allotter

1. Invigilators want favorite city: Goa?
2. Centres want experienced invigilators
3. Allotter: No goof-ups in exam: top priority

2nd allocation Problem: invigilator/centre

GATE: national level competitive exam: critical for IITs

Two sets: invigilators $\mathbf{I}$ and examination centres C: allot $\mathbf{I}$ to C

1. Data: invigilators give preference cities.

   Some invigilators are more experienced/senior.

2. Centre: number of invigilators required

   Centres have different experience

Three parties: invigilators, centres, Allotter

1. Invigilators want favorite city: Goa?
2. Centres want experienced invigilators
3. Allotter: No goof-ups in exam: top priority

2nd allocation Problem: invigilator/centre

GATE: national level competitive exam: critical for IITs

Two sets: invigilators $\mathbf{I}$ and examination centres C: allot $\mathbf{I}$ to C

1. Data: invigilators give preference cities.

   Some invigilators are more experienced/senior.

2. Centre: number of invigilators required

   Centres have different experience

Three parties: invigilators, centres, Allotter

1. Invigilators want favorite city: Goa?
2. Centres want experienced invigilators
3. Allotter: No goof-ups in exam: top priority

2nd allocation Problem: invigilator/centre

GATE: national level competitive exam: critical for IITs

Two sets: invigilators $\mathbf{I}$ and examination centres C: allot $\mathbf{I}$ to C

1. Data: invigilators give preference cities.

   Some invigilators are more experienced/senior.

2. Centre: number of invigilators required

   Centres have different experience

Three parties: invigilators, centres, Allotter

1. Invigilators want favorite city: Goa?
2. Centres want experienced invigilators
3. Allotter: No goof-ups in exam: top priority

2nd allocation Problem: invigilator/centre

GATE: national level competitive exam: critical for IITs

Two sets: invigilators $\mathbf{I}$ and examination centres C: allot $\mathbf{I}$ to C

1. Data: invigilators give preference cities.

   Some invigilators are more experienced/senior.

2. Centre: number of invigilators required

   Centres have different experience

Three parties: invigilators, centres, Allotter

1. Invigilators want favorite city: Goa?
2. Centres want experienced invigilators
3. Allotter: No goof-ups in exam: top priority

Allotter's requirement

Exam conduct smooth!

1. Equalize experience within centre
2. Balance: Centre experience + invigilator experiences
3. Balance work across people and weekends
    1. Allot senior invigilator's higher choices
    2. Minimize average choice number in an allotment

Allotment method

1. Construct weights carefully: student-score (seniority), preferences

   Maximum weight matching: Hungarian method:

   - Munkres in Python
   - max_weight_matching using NetworkX

2. Markov-Chain-Monte-Carlo (MCMC):

   pymc in Python

Allotment: outside scope of today's talk

Today: data analysis, using Pandas

Why analyze data: just allot?

Allotter's dilemma

1. Is an allotment good?
2. Choices perhaps skewed
3. Seek more choices?

Data decides how good an allotment can be

Typical questions:

1. Did people give sufficiently different choices?
2. Some course (city) too popular, some course (city) unwanted
3. 1st weekend for GATE more sought than 2nd weekend (for 2016)
4. If data is skewed, allotment inevitably bad
5. Allot some manually: critical courses/cities
6. Quickly extract specific type of choosers
7. Obtain 'breakups': pie-charts, bar-diagrams

Why Pandas?

Pandas: simplifies data processing tasks:

1. Reading and writing files
2. Data extraction
3. Data manipulations
4. Getting statistics on data
5. Plotting and visualization