Foundations of Data Science

Master in Data Science

2022 / 2023

Data Sources & Formats

Data sources

- Bulk download locally
- Read from database
- API
- Web scraping

Bulk data formats

- CSV, TSV
- JSON
- XML
- XLS
- Text, image, maps, …

- Big Data file formats
  - Parquet: column based, binary, metadata, efficient read of only some columns
  - Avro: row based, binary w/ JSON schema, efficient split

Web APIs

- Data can be accessed through Applied Programming Interfaces (APIs)

https://api.v2.emissions-api.org/api/v2/carbonmonoxide/average.json?country=PT&begin=2021-01-01&end=2021-01-31


```
[
  {
    "average": 0.03358262626731649,
    "end": "2021-01-01T15:29:39.374000Z",
    "start": "2021-01-01T15:28:13.695000Z"
  },
  {
    "average": 0.0348591667345979,
    "end": "2021-01-02T13:30:44.980000Z",
    "start": "2021-01-02T13:29:52.902000Z"
  },
  {
    "average": 0.03303938828563938,
    "end": "2021-01-03T14:51:35.161000Z",
    "start": "2021-01-03T14:50:13.683000Z"
  },
```

Web APIs

- Data can be accessed through Applied Programming Interfaces (APIs)

https://api.v2.emissions-api.org/api/v2/carbonmonoxide/average.json?country=PT&begin=2021-01-01&end=2021-01-31

```
import urllib.request, json

u = 'https://api.v2.emissions-api.org/api/v2/carbonmonoxide/average.json?country=PT&begin=2021-01-01&end=2021-01-31'
with urllib.request.urlopen(u) as url:
    data = json.loads(url.read().decode())
    print(data[0])

{
  'average': 0.03358262626731649,
  'end': '2021-01-01T15:29:39.374000Z',
  'start': '2021-01-01T15:28:13.695000Z'
}
```

Scraping

```
import requests
from bs4 import BeautifulSoup

r = requests.get('https://www.dn.pt')
root = BeautifulSoup(r.content)
headlines = root.find_all("header", {"class": "t-am-head"})
for h in headlines:
    if h.find("h2"): print(h.find("h2").text)
```

```
De Bragança ao Algarve há hospitais em "estado de calamidade"
Assista ao segundo dia do Portugal Mobi Summit
Quando português é sinónimo de doce
Revisão de leis laborais avança. Mexida nas horas extra ainda em aberto
Nova vaga alastra na Europa sobretudo onde há menos vacinação
Ligações da Transtejo com supressão de carreiras a partir das 13:30
```

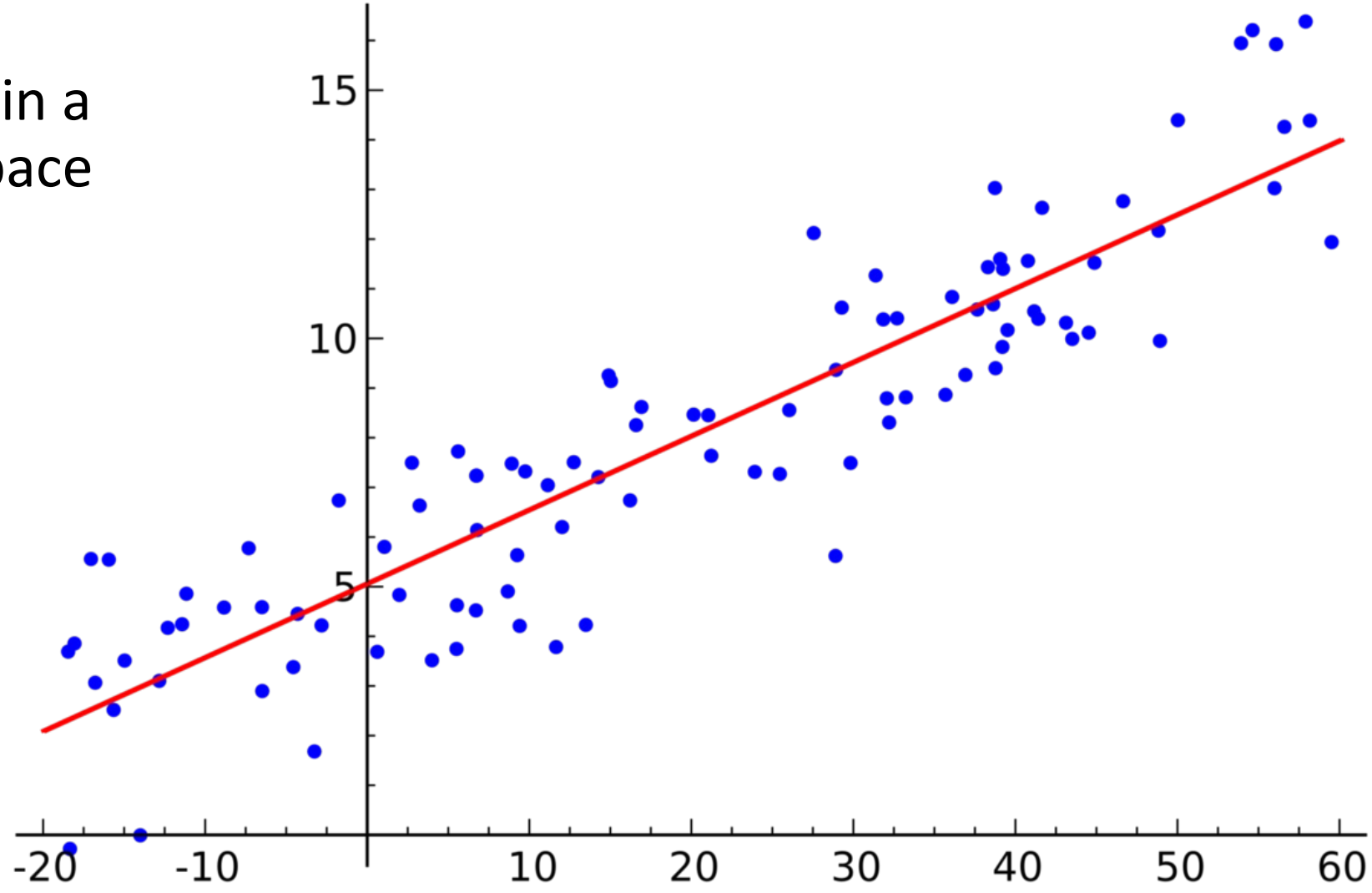
Scrapering: Spydering + Scraping

```
import scrapy


class QuotesSpider(scrapy.Spider):
    name = 'quotes'
    start_urls = [
        'http://quotes.toscrape.com/tag/humor/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'author': quote.xpath('span/small/text()').get(),
                'text': quote.css('span.text::text').get(),
            }

        next_page = response.css('li.next a::attr("href")').get()
        if next_page is not None:
            yield response.follow(next_page, self.parse)
```

Data Representation & Manipulation

Data

- Generally points in a *d*-dimensional space (here *d*=2)



Data manipulation

- Data Science == manipulating and computing on data
  - Large to very large, but somewhat “structured” data

Comp Science / Programming		Data Science
Imperative code to manipulate data structures	→	Sequences/pipelines of operations on data

Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data

One-dimensional Arrays, Vectors

0.1	2	3.2	6.5	3.4	4.1

“data”	”representation”	”i.e.”

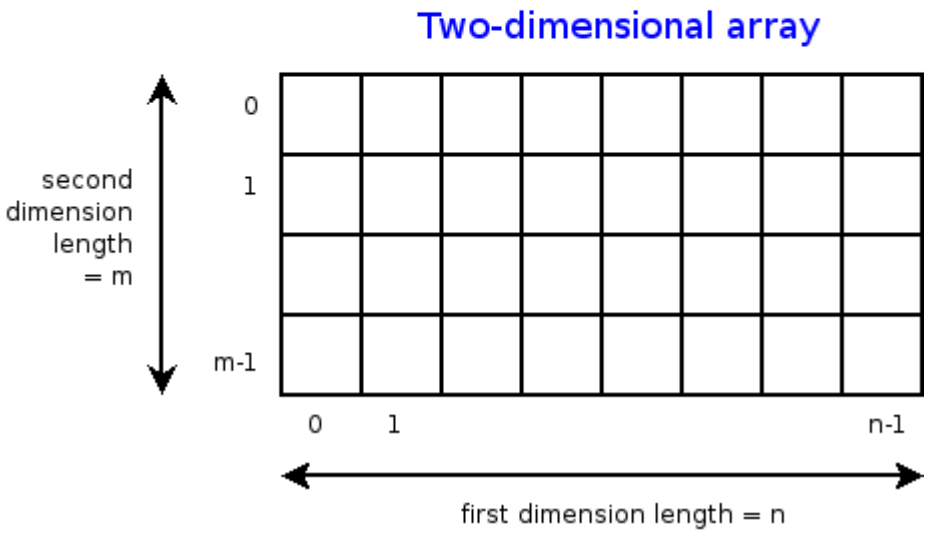
Indexing
Slicing/subsetting
Filter
‘map’ → apply a function to every element
’reduce/aggregate’ → combine values to get a single scalar (e.g., sum, median)

Given two arrays: **Dot and cross products**

2. Data Operations: take one or more datasets as input and produce one or more datasets as output

Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data



Indexing
Slicing/subsetting
Filter
'map' → apply a function to every element
'reduce/aggregate' → combine values to get a single scalar (e.g., sum, median)

Given two arrays: **Dot and cross products**

2. Data Operations: take one or more datasets as input and produce one or more datasets as output

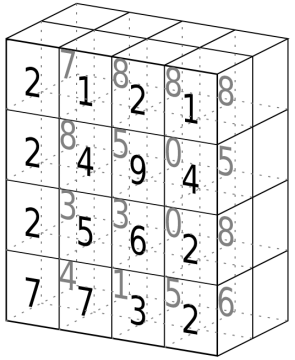
Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data

Matrices, Tensors

3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

tensor of dimensions [6,4]
(matrix 6 by 4)



tensor of dimensions [4,4,2]

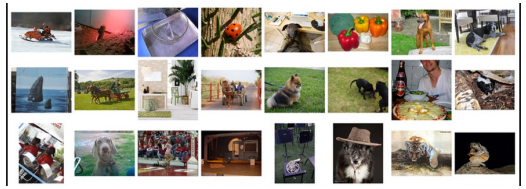
n-dimensional array operations
+
Linear Algebra
- **Matrix/tensor multiplication**
- **Transpose**
- **Matrix-vector multiplication**
- **Matrix factorization**

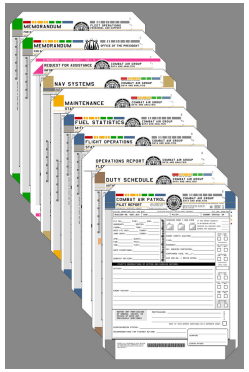
2. Data Operations: take one or more datasets as input and produce one or more datasets as output

Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data

Sets: of Objects





Filter
Map
Union

Reduce/Aggregate

Sets: of (Key, Value Pairs)

(amol@cs.umd.edu,(email1, email2,…))

(john@cs.umd.edu,(email3, email4,…))

Given two sets, **Combine/Join** using "keys"

Group and then aggregate

2. Data Operations: take one or more datasets as input and produce one or more datasets as output

Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data

Tables/Relations == Sets of Tuples

company	division	sector	tryint
00nil_Combined_Company	00nil_Combined_Division	00nil_Combined_Sector	14625
apple	00nil_Combined_Division	00nil_Combined_Sector	10125
apple	hardware	00nil_Combined_Sector	4500
apple	hardware	business	1350
apple	hardware	consumer	3150
apple	software	00nil_Combined_Sector	5625
apple	software	business	4950
apple	software	consumer	675
microsoft	00nil_Combined_Division	00nil_Combined_Sector	4500
microsoft	hardware	00nil_Combined_Sector	1890
microsoft	hardware	business	855
microsoft	hardware	consumer	1035
microsoft	software	00nil_Combined_Sector	2610
microsoft	software	business	1215
microsoft	software	consumer	1395

Filter rows or columns

”Join” two or more relations

”Group” and “aggregate” them

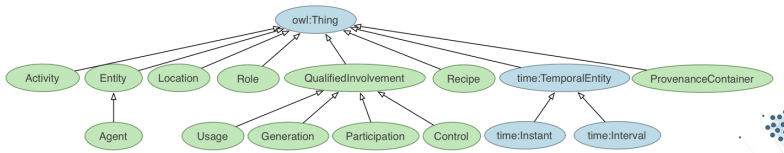
Relational Algebra formalizes some of them

2. Data Operations: take one or more datasets as input and produce one or more datasets as output

Data Manipulation and Computation

1. Data Representation: what is the natural way to think about given data

Trees/Graphs



”Path” queries

Graph Algorithms and Transformations

Network Science

2. Data Operations: take one or more datasets as input and produce one or more datasets as output

Table Operations

Tables

Special Column, called “Index”, or “ID”, or “Key”
Usually, no duplicates Allowed

Variables
(also called Attributes, or Columns, or Labels)

Observations, Rows, or Tuples

ID	age	wgt_kg	hgt_cm
1	12.2	42.3	145.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

Tables

ID	Address
1	College Park, MD, 20742
2	Washington, DC, 20001
3	Silver Spring, MD 20901

ID	age	wgt_kg	hgt_cm
1	12.2	42.3	145.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085

1. Select/slicing

- Select only some of the rows, or some of the columns, or a combination

ID	age	wgt_kg	hgt_cm
1	12.2	42.3	145.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

Only columns ID and Age

ID	age
1	12.2
2	11.0
3	15.6
4	35.1

Only rows with wgt > 41

ID	age	wgt_kg	hgt_cm
1	12.2	42.3	145.1
3	15.6	65.3	165.3
4	35.1	84.2	185.8

Both

ID	age
1	12.2
3	15.6
4	35.1

2. Aggregate/Reduce

- Combine values across a column into a single value

ID	age	wgt_kg	hgt_cm
1	12.2	42.3	145.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

SUM → **73.9 232.6 640.0**

MAX → **35.1 84.2 185.8**

SUM(wgt_kg^2 - hgt_cm) → **14167.66**

What about ID/Index column?

Usually not meaningful to aggregate across it

May need to explicitly add an ID column

3. Map

- Apply a function to every row, possibly creating more or fewer columns

ID	Address
1	College Park, MD, 20742
2	Washington, DC, 20001
3	Silver Spring, MD 20901

→

ID	City	State	Zipcode
1	College Park	MD	20742
2	Washington	DC	20001
3	Silver Spring	MD	20901

Variations that allow one row to generate multiple rows in the output (sometimes called “flatmap”)

4. Group By

- Group tuples together by column/dimension

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

By ‘A’ →

A = foo

ID	B	C
1	3	6.6
3	4	3.1
4	3	8.0
7	4	2.3
8	3	8.0

A = bar

ID	B	C
2	2	4.7
5	1	1.2
6	2	2.5

4. Group By

- Group tuples together by column/dimension

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

By ‘B’ →

B = 1

ID	A	C
5	bar	1.2

B = 2

ID	A	C
2	bar	4.7
6	bar	2.5

B = 3

ID	A	C
1	foo	6.6
4	foo	8.0
8	foo	8.0

B = 4

ID	A	C
3	foo	3.1
7	foo	2.3

4. Group By

- Group tuples together by column/dimension

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

By ‘A’, ‘B’ →

A = bar, B = 1

ID	C
5	1.2

A = bar, B = 2

ID	C
2	4.7
6	2.5

A = foo, B = 3

ID	C
1	6.6
4	8.0
8	8.0

A = foo, B = 4

ID	C
3	3.1
7	2.3

5. Group By Aggregate

- Compute one aggregate per group

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

Group by 'B'
Sum on C

B = 1

ID	A	C
5	bar	1.2

B = 1

Sum (C)
1.2

B = 2

ID	A	C
2	bar	4.7
6	bar	2.5

B = 2

Sum (C)
7.2

B = 3

ID	A	C
1	foo	6.6
4	foo	8.0
8	foo	8.0

B = 3

Sum (C)
22.6

B = 4

ID	A	C
3	foo	3.1
7	foo	2.3

B = 4

Sum (C)
5.4

5. Group By Aggregate

- Final result usually seen as a table

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

Group by 'B'
Sum on C

B = 1

Sum (C)
1.2

B = 2

Sum (C)
7.2

B = 3

Sum (C)
22.6

B = 4

Sum (C)
5.4

B	SUM(C)
1	1.2
2	7.2
3	22.6
4	5.4

6. Union/Intersection/Difference

- Set operations – only if the two tables have identical attributes/columns

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0

U

ID	A	B	C
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

→

ID	A	B	C
1	foo	3	6.6
2	bar	2	4.7
3	foo	4	3.1
4	foo	3	8.0
5	bar	1	1.2
6	bar	2	2.5
7	foo	4	2.3
8	foo	3	8.0

7. Merge or Join

- Combine rows/tuples across two tables if they have the same key

ID	A	B
1	foo	3
2	bar	2
3	foo	4
4	foo	3

$\bowtie$

ID	C
1	1.2
2	2.5
3	2.3
5	8.0

$\longrightarrow$

ID	A	B	C
1	foo	3	1.2
2	bar	2	2.5
3	foo	4	2.3

What about IDs not present in both tables?

Often need to keep them around

Can "pad" with NaN

7. Merge or Join

- Combine rows/tuples across two tables if they have the same key
- Outer joins can be used to ”pad” IDs that don’t appear in both tables
- Three variants: LEFT, RIGHT, FULL
- SQL Terminology -- Pandas has these operations as well

ID	A	B
1	foo	3
2	bar	2
3	foo	4
4	foo	3

⋈

ID	C
1	1.2
2	2.5
3	2.3
5	8.0

→

ID	A	B	C
1	foo	3	1.2
2	bar	2	2.5
3	foo	4	2.3
4	foo	3	NaN
5	NaN	NaN	8.0