REPRESENTING DATA

Practical Lectures

by Chris Emmery (MSc)

@_cmry • @cmry

LAST WEEK

- Data
- Features
- Algorithms

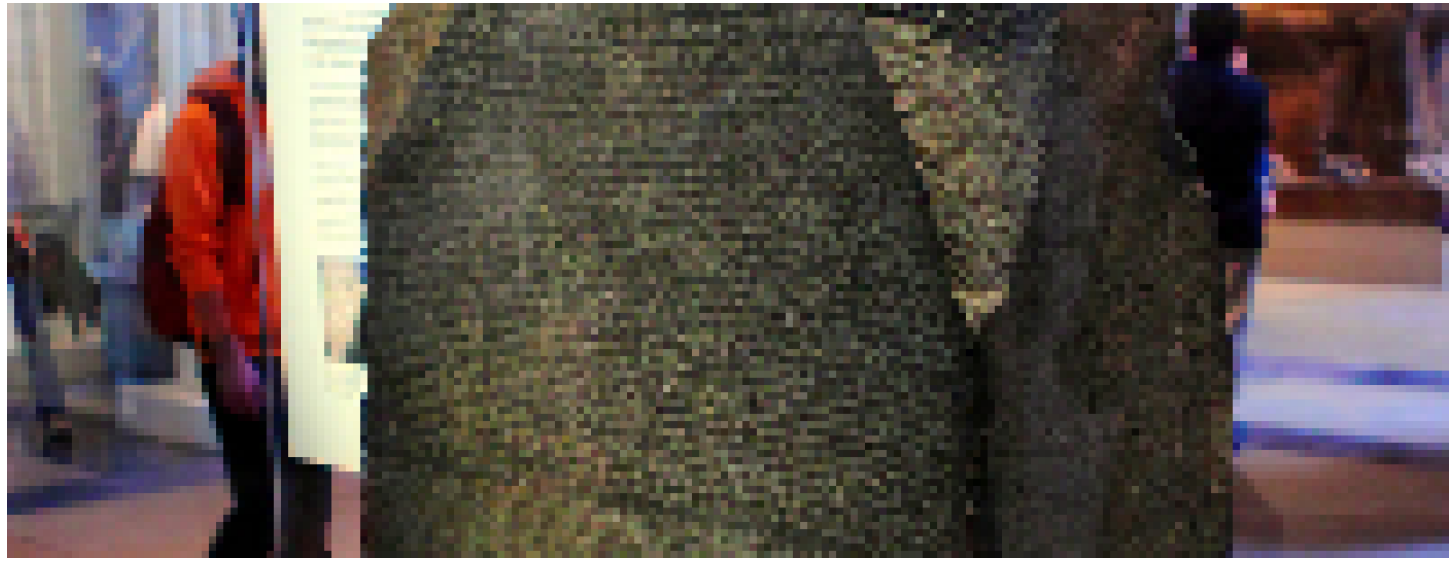
THIS WEEK

- Data ←
- Features
- Algorithms

HOW DO WE GET DATA?

- Pre-mades: e.g. Kaggle, UCI, Snap.
- Dumps: e.g. IMDB, Reddit, MovieLens.
- Scientific repositories: e.g. dataverse.
- (Web) API's: e.g. Twitter, Reddit.
- Web scraping.
- At industry-level: databases.

FILE FORMATS



HOW IS DATA FORMATTED?

- **CSV** → flat (table format), named columns and rows. Separators and quotes.
- **JSON** → hierchical, lists and key/values. Widely used for API's and document-based databases.
- **XML** → hierchical, tags to name items. Very common standard; easy to evaluate if according to some pre-defined structure.

CSV - OBJECT

(for sale)

CSV - FILE

```
CRIM,ZN,INDUS,CHAS,NOX,RM,AGE,DIS,RAD,TAX,PTRATIO,B,LSTAT,MEDV
0.00632,18.00,2.310,0,0.5380,6.5750,65.20,4.0900,1,296.0,15.30,396.90,4.98,2
0.02731,0.00,7.070,0,0.4690,6.4210,78.90,4.9671,2,242.0,17.80,396.90,9.14,21
0.02729,0.00,7.070,0,0.4690,7.1850,61.10,4.9671,2,242.0,17.80,392.83,4.03,34
0.03237,0.00,2.180,0,0.4580,6.9980,45.80,6.0622,3,222.0,18.70,394.63,2.94,33
```

CRIM	ZN	INDUS	CHAS	NOX	...
0.00632	18.00	2.310	0	0.5380	...
0.02731	0.00	7.070	0	0.4690	...
0.02729	0.00	7.070	0	0.4690	...
0.03237	0.00	2.180	0	0.4580	...

JSON - OBJECT


SitePoint JavaScript
@SitePointJS
Follow
Creating a Grocery List Manager Using Angular, Part 1: Add & Display Items buff.ly /2sr60pf #Angular
Creating a Grocery List Manager Using Angular, Part 1: Ad...
An Angular application is made up of components. In an Angular application, a component consists of an HTML template and a component class. From the official docs:Components are the
code.tutsplus.com
11:00 PM - 22 Jun 2017
1 Retweet 1 Like
1
1

JSON - FILE


```
[{
  "created_at": "Thu Jun 22 21:00:00 +0000 2017",
  "id": 877994604561387500,
  "id_str": "8779946045613875 20",
  "text": "Creating a Grocery List Manager Using Angular, Part 1: Add & Disp
  "truncated": false,
  "entities": {
    "hashtags": [{
      "text": "Angular",
      "indices": [103, 111]
    }],
    "symbols": [],
    "user_mentions": [],
    "urls": [{
      "url": "https://t.co/xFox78juL1",
      "expanded_url": "http://buff.ly/2sr60pf",
      "display_url": "buff.ly/2sr60pf",
      "indices": [79, 102]
    }]
  },
  "user": {
    "id": 772682964,
    "id_str": "772682964"
```

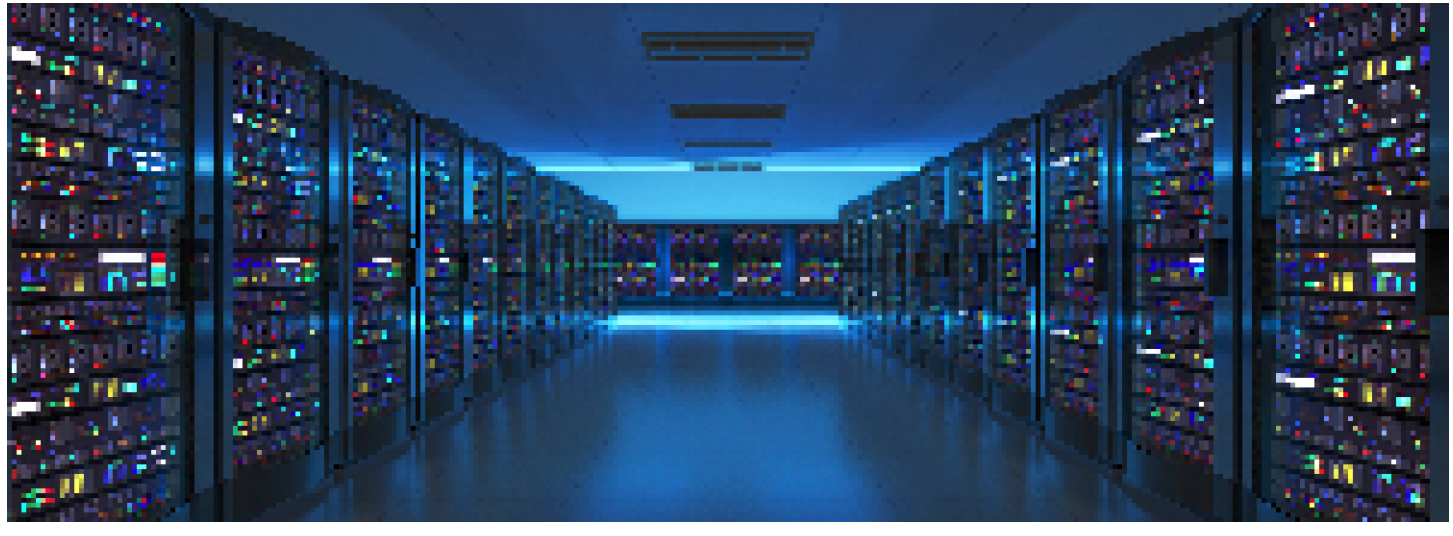

```
tweet['text'] = "Creating a Grocery List Manager ..."
tweet['entities']['hashtags'][0]['text'] = "Angular"
tweet['user']['screen_name'] = "SitePointJS"
```

XML VS JSON

```xml
<!--?xml version="1.0"?-->
<book id="123">
  <title>Object Thinking</title>
  <author>David West</author>
  <published>
    <by>Microsoft Press</by>
    <year>2004</year>
  </published>
</book>
```

```json
{
  "id": 123,
  "title": "Object Thinking",
  "author": "David West",
  "published": {
    "by": "Microsoft Press",
    "year": 2004
  }
}
```

DATABASES



WHAT ARE DATABASES?

(NON) RELATIONAL

- **SQL**: e.g. MySQL, PostgreSQL, SQLite, MariaDB.
  - Pre-defined, structured, relational → not easy to scale horizontally, but robust, and well-supported.
- **NoSQL**: e.g. MongoDB, CouchDB, Cassandra, Redis.
  - Flexible, multiple designs, not (only) relational → automatic and easy scalability, many different varieties, upcoming thus less well supported.

SQL

```
CREATE TABLE STATION
(ID INTEGER PRIMARY KEY,
CITY CHAR(20),
STATE CHAR(2),
LAT_N REAL,
LONG_W REAL);
```

```
INSERT INTO STATION VALUES (13, 'Phoenix', 'AZ', 33, 112);
INSERT INTO STATION VALUES (44, 'Denver', 'CO', 40, 105);
INSERT INTO STATION VALUES (66, 'Caribou', 'ME', 47, 68);
```

```
SELECT CITY, STATE FROM STATION WHERE LAT_N > 39.7;
```

CITY	STATE
Denver	CO
Caribou	ME

NOSQL

```
db.station.insertMany([
  {"id": 13,
   "city": "Phoenix", "state": "AZ",
   "lat_n": 33, "long_w": 112},
  {"id": 44,
   "city": "Denver", "state": "CO",
   "lat_n": 40, "long_w": 105},
  {"id": 66,
   "city": "Caribou", "state": "ME",
   "lat_n": 47, "long_w": 68}
])
```

```
db.station.find({"lat_n": {$gt: 39.7}},
                {"city": 1, "state": 1})
```

CITY	STATE
Denver	CO
Caribou	ME

REPRESENTING DATA AND KNOWLEDGE

ANALYSIS VS. PREDICTIVE ANALYSIS

- 80% analysis: collecting, discovering, cleaning and aggregating data from different databases / sources to generate creative insights.
- 20% prediction: use all the above to generate insights only possible by machines.

EXAMPLE OF MANUAL ANALYSIS

What's in a Tweet?

- Text (hashtags, urls, mentions, emojis)
- Users (images, names, descriptions, links)
- Time
- Location
- Networks (reach, spread)

All directly in human readable data.

HOW DO WE REPRESENT DATA FOR MACHINES?

rating	actor	year	label
6.5	johansson	2017	okay
7	pitt	2013	good
8	doubleday	2015	very good

REPRESENTATION: REGRESSION

$$X = \begin{bmatrix} 1 & 2016 & 3 \\ 0 & 2014 & 4 \\ 6 & 2012 & 5 \end{bmatrix}$$

$$Y = \begin{bmatrix} 6.5 \\ 7.0 \\ 8.0 \end{bmatrix}$$

REPRESENTATION: CLASSIFICATION

$$X = \begin{bmatrix} 6.5 & 1 & 2016 \\ 7.0 & 0 & 2014 \\ 8.0 & 6 & 2012 \end{bmatrix}$$

$$Y = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$$

VECTOR SPACES

$$\overrightarrow{x_1} = \langle 6.5, 1, 2016 \rangle$$

$$\overrightarrow{x_2} = \langle 7.0, 0, 2014 \rangle$$

$$\overrightarrow{x_3} = \langle 8.0, 6, 2012 \rangle$$