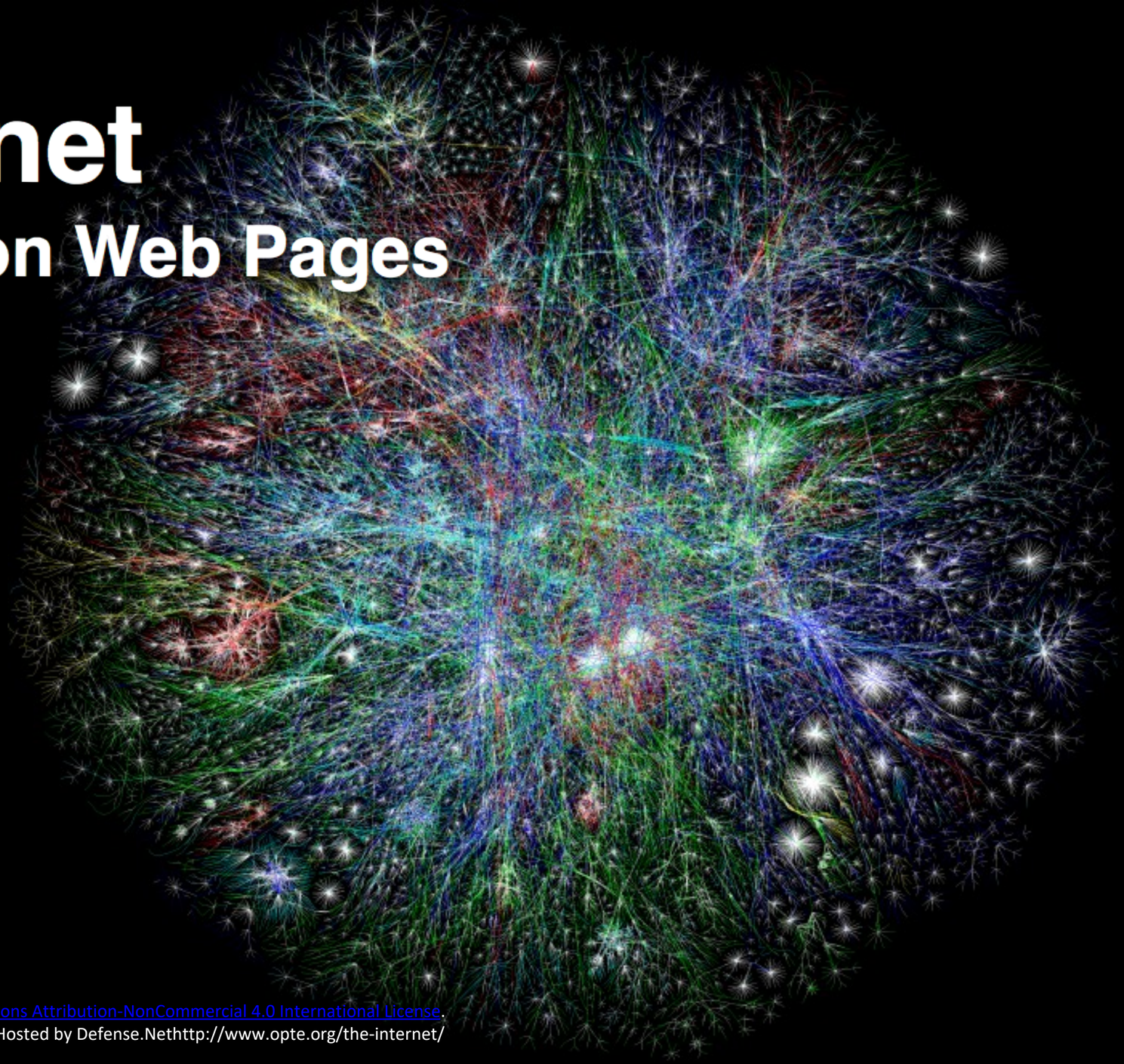
DLI Accelerated Data Science Teaching Kit

Lecture 17.1 - How to Represent and Store Graphs

Internet

50 Billion Web Pages



Facebook

2+ Billion Users

Many More

Twitter

Who-follows-whom (500 million users)

Amazon

Who-buys-what (120 million users)

Cellphone network

Who-calls-whom (100 million users)

Protein-protein interactions

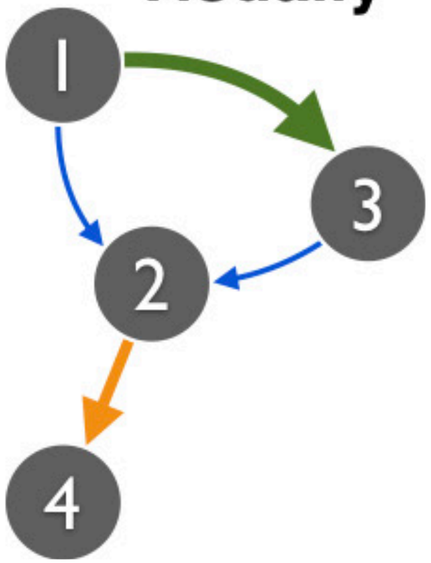
200 million possible interactions in human genome

How to Represent a Graph?

Conceptually.
Visually.
Programmatically.

How to Represent a Graph?

Visually



Adjacency matrix

Target node

Source node	1	2	3	4
1	0	1	3	0
2	0	0	0	2
3	0	1	0	0
4	0	0	0	0

Adjacency list

1: 2, 3
2: 4
3: 2

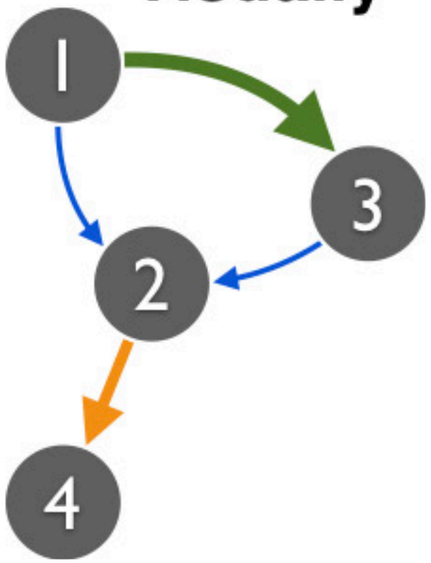
Edge list

1, 2, 1
1, 3, 3
2, 4, 2
3, 2, 1

- Most common distribution format
- Can be painful to parse when edges/nodes have many attributes/columns (e.g., text with quotes)

How to Represent a Graph?

Visually



Adjacency matrix

Target node

Source node	1	2	3	4
1	0	1	3	0
2	0	0	0	2
3	0	1	0	0
4	0	0	0	0

Adjacency list

1: 2, 3
2: 4
3: 2

Edge list

1, 2, 1
1, 3, 3
2, 4, 2
3, 2, 1

Each node is often identified by a numeric ID. Why?

Assigning an ID to a Node

- Use a “map” (Java) / “dictionary” (Python) / SQLite
- Same concept: given an entity/node (e.g., “Tom”) not seen before, assign a number to it

Assigning Node IDs using SQLite

Create an index for “name”. Then write a join query.

rowid	name
1	Tom
2	Sandy
3	Richard
4	Polo

Rowid is a hidden column automatically created by SQLite

source	target
Tom	Sandy
Polo	Richard

source	target
1	2
4	3

How to Store (Large) Graph?

On your laptop computer

- SQLite
- Neo4j (GPL license)
  http://neo4j.com/licensing/

On a server

- MySQL, PostgreSQL, etc.
- Neo4j (?)

With a cluster of machines

- Titan (on top of HBase), S2Graph — if you need real time read and write
- Hadoop (generic framework) — if batch processing is fine
- Hama, Giraph, inspired by Google's Pregel
- FlockDB, by Twitter

Storing Large Graphs

My research group like to use SQLite. Why? Great for our use cases.

- Easily handle up to gigabytes ~ tens of millions of nodes/edges!
- Very easy to maintain: one cross-platform file (even works on iPad)
- APIs in many languages
- Queries are easy! e.g., find all nodes' degrees = 1 SQL statement
- Bonus: SQLite supports full-text search

SQLite Graph Database Schema

Simplest schema:

```
edges(source_id, target_id)
```

More sophisticated (flexible; lets you store more things):

```
CREATE TABLE nodes (
    id INTEGER PRIMARY KEY,
    type INTEGER DEFAULT 0,
    name VARCHAR DEFAULT '');

CREATE TABLE edges (
    source_id INTEGER,
    target_id INTEGER,
    type INTEGER DEFAULT 0,
    weight FLOAT DEFAULT 1,
    timestamp INTEGER DEFAULT 0,
    PRIMARY KEY(source_id, target_id, timestamp));
```

DEEP
LEARNING
INSTITUTE

PRAIRIE VIEW
A&M UNIVERSITY

DLI Accelerated Data Science Teaching Kit

Thank You