Graph-Based Analysis of Chlamydia Spread

DATASCI 205 | Section 9 | Spring 2025

Azariah, Solomon | Hou, Yiwen | Karpman, Jonah | Nap, Ronald

Agenda

Business Case Scenario

Dataset Overview

Graph Design & Construction

Graph Algorithms

MongoDB & Redis Use Cases

Concluding Thoughts

Business Case Scenario: Early Detection & Regional Coordination

Problem : Public health officials need to identify outbreaks early to coordinate response

Impact : Faster detection enables quicker resource deployment and reduces disease spread

Goal : Track regional trends in chlamydia infections to detect clusters and respond quickly

Why Graphs : Outbreak patterns are often spatial, temporal, and relational — graphs reveal these connections better than relational databases

Dataset Overview: CDC NNDSS

Source : CDC National Notifiable Diseases Surveillance System (NNDSS)

Scope : Weekly reports of 100+ nationally notifiable diseases across U.S. states and territories (2022–2025)

Focus disease : Chlamydia trachomatis

Filtered records : 6594 rows

Processing steps:

1. Retrieved full dataset via CDC's public API
2. Cleaned and standardized fields
3. Filtered out regional aggregates and non-reporting rows
4. Focused on a single disease
5. Calculated weekly case differences by region

Graph Design & Construction: Neo4j

Nodes:

- Location (e.g., California, Texas)
- Report (weekly report of a specific disease in a location)

Relationships:

- (:Location)-[:CONTAINS]->(:Report)
  - Links a location to its corresponding weekly disease report
- (:Report)-[:NEXT]->(:Report)
  - Temporal sequence of reports in the same location (e.g., week 1 → week 2)
- (:Location)-[:BORDERS]->(:Location)
  - Indicates neighboring regions to support geographic proximity in analysis

Graph Algorithm:

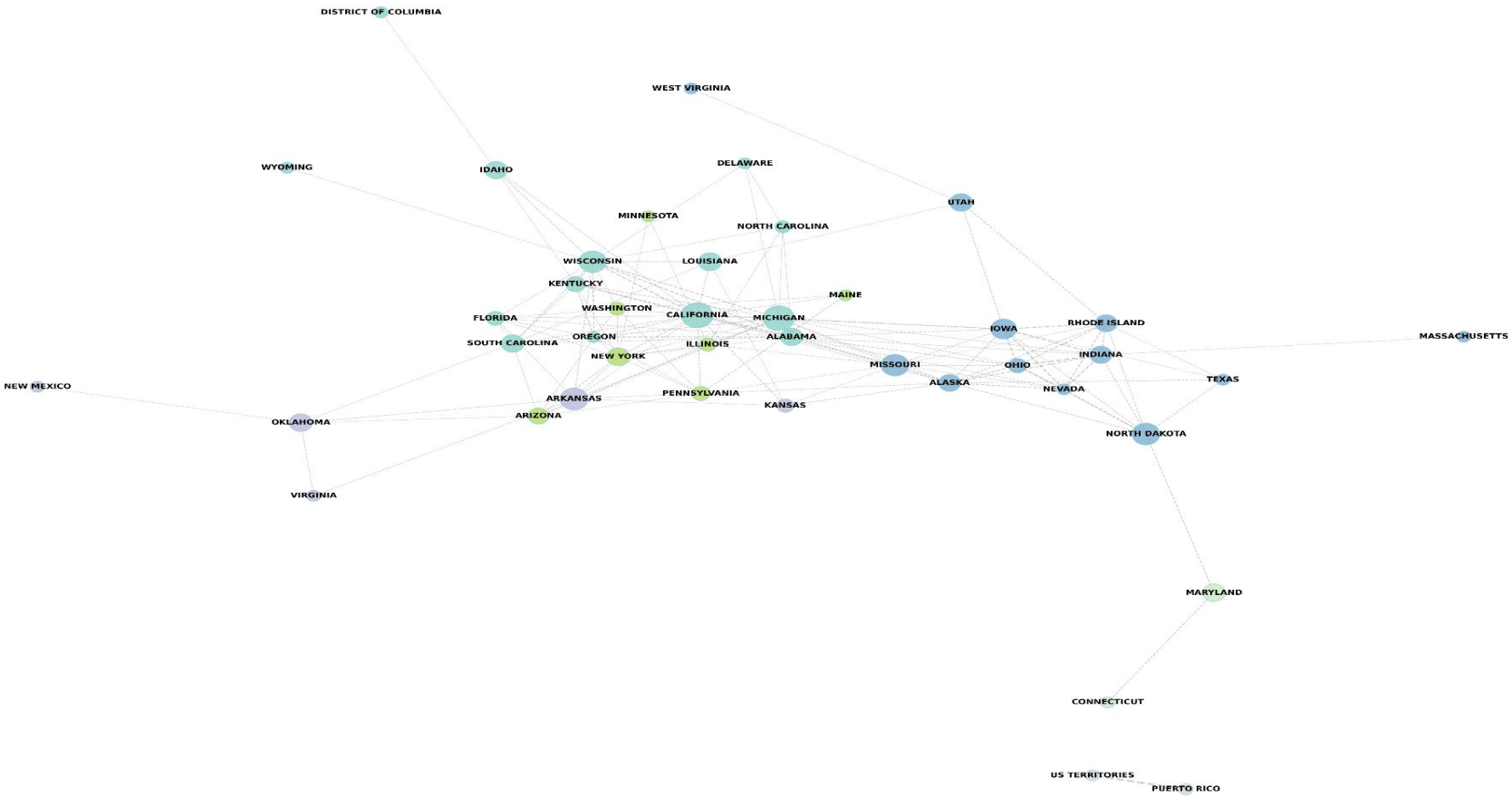
Louvain (Community Detection)

- **Goal**: Group states with **similar infection** patterns (rise/fall curves).
- **Discovery** : We found multiple clusters (colors). Some states form a tight Southeastern block, others are scattered across the country but still share **synchronized** trends.
-

Betweenness Centrality (Bridge States)

- **Goal**: Identify “choke points” that lie on **shortest paths** in the infection network.
- **Discovery** : Certain **moderate** -incidence states (like California) act as **bridges** connecting clusters. Stopping infection there slows cross-cluster spread.

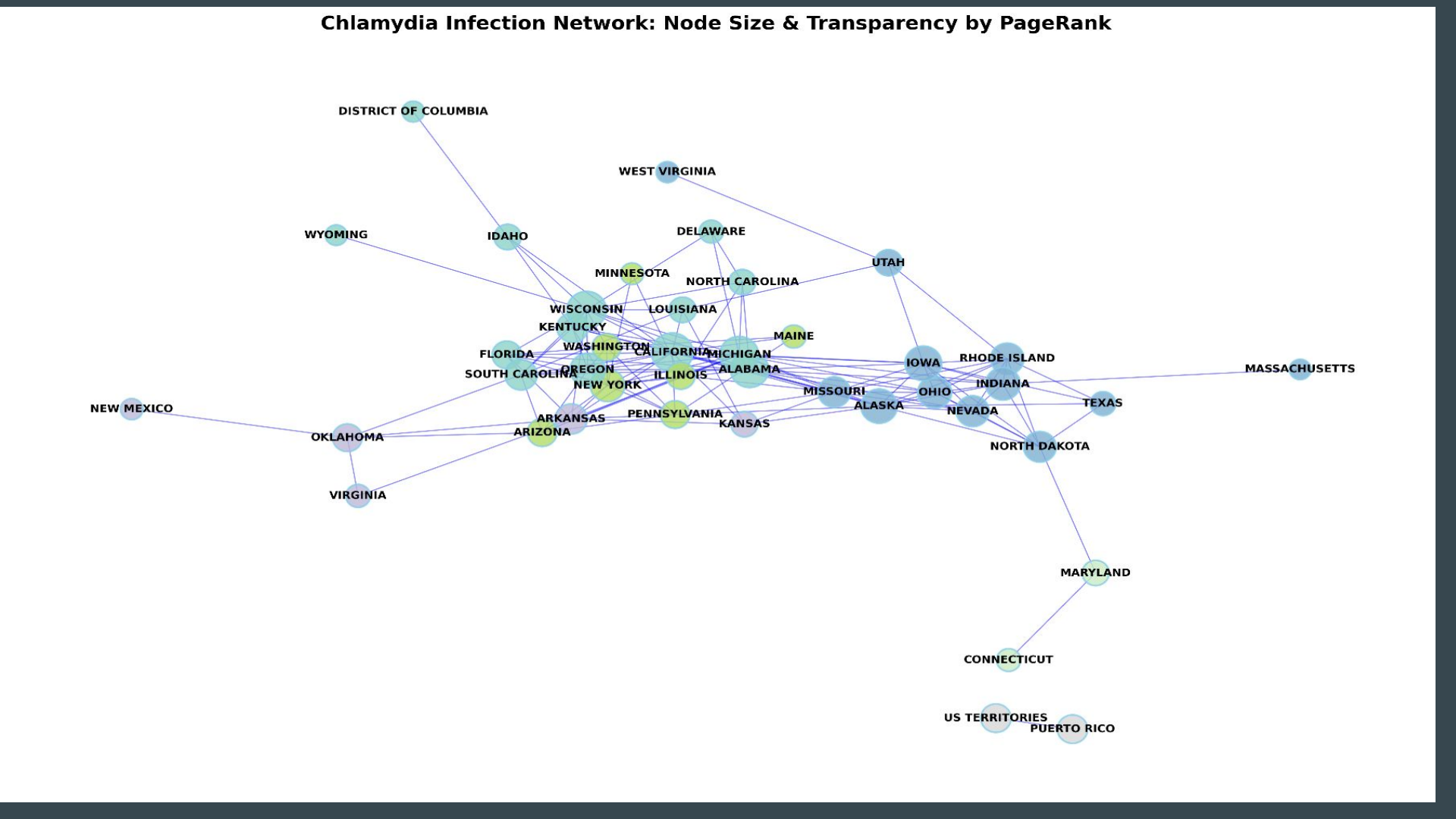
Network of Similar Chlamydia Infection Patterns (Louvain Communities & Betweenness Centrality)



Graph Algorithm:

PageRank (Hubs)

- **Goal**: Pinpoint states that are “well connected to other well-connected states” (influence).
- **Discovery** : High-population states often have **partial** alignment with many sub-groups, making them “hubs.” Even if not top in raw incidence, they can **amplify** infection waves across multiple clusters.


Chlamydia Infection Network: Node Size & Transparency by PageRank
DISTRICT OF COLUMBIA
WEST VIRGINIA
WYOMING
IDAHO
DELAWARE
UTAH
MINNESOTA
NORTH CAROLINA
WISCONSIN
LOUISIANA
KENTUCKY
MAINE
WASHINGTON
CALIFORNIA
MICHIGAN
FLORIDA
IOWA
RHODE ISLAND
MASSACHUSETTS
SOUTH CAROLINA
OREGON
ILLINOIS
ALABAMA
INDIANA
NEW YORK
MISSOURI
OHIO
NEW MEXICO
ALASKA
TEXAS
ARKANSAS
PENNSYLVANIA
NEVADA
KANSAS
ARIZONA
OKLAHOMA
NORTH DAKOTA
VIRGINIA
MARYLAND
CONNECTICUT
US TERRITORIES
PUERTO RICO

Graph Algorithm:

Complex, Iterative Graph Algorithms

- **Louvain** needs repeated merging to maximize modularity.
- **Betweenness** requires shortest-path computations.
- **PageRank** is iterative.

Native Graph Queries

- In Neo4j + GDS, we can directly store states as nodes, infection correlation as edges with weights, then run built-in Louvain, Betweenness, PageRank.
- Relational: Lacks an efficient, built-in way to do multi-step graph traversals or modularity calculations for community detection.

Visual & Conceptual Simplicity

- Graph data is intuitive for infection correlation—states are nodes, correlation is an edge. We can easily visualize bridging states/clusters.
- Relational queries hide these relationships in multiple join tables, complicating iterative exploration and real-time visualization.

Graph Findings & Why We Used Infection Rate Correlations

Key Findings

- Clusters, not always geographic: Louvain revealed multiple color-coded communities where states share similar infection curves. Some clusters include states far apart physically, proving *non*-adjacent states can co-fluctuate.
- Bridging States: Betweenness shows moderate-incidence states (e.g. California) connect multiple communities, potentially channeling outbreaks across regions if not monitored.
- Hubs vs. High Incidence: PageRank identified states with strong ties to other well-connected nodes, emphasizing that being influential in the network doesn't always mean topping raw case counts.

Why Infection Patterns Instead of Geography?

- Disease Ignores Physical Borders: States can share risk factors, travel corridors, or demographic overlaps even if thousands of miles apart.
- Realistic Spread View: Correlation of actual infection timelines (i.e. *who peaks with whom*) matters more for outbreak forecasting and multi-state collaboration than adjacency.
- Actionable: If State A (distant) consistently syncs with State B's infection surges, they should coordinate interventions—something a purely geographic approach wouldn't highlight.

MongoDB Use Case: Flexible Outbreak Summaries

Benefits of MongoDB

- Store pre-aggregated outbreak metrics for each region and week
- Easily adapt to new fields or summary types over time
- Power quick lookups and visualizations for dashboards

Why not relational?

- Requires multiple joins to compute summaries
- Schema changes are rigid and time-consuming
- Slower performance for dynamic queries

Example MongoDB document


```
{
 "reporting_area": "CALIFORNIA",
  "year": 2024,
  "week": 13,
  "disease": "CHLAMYDIA TRACHOMATIS",
  "cases_current_week": 1253,
  "cases_52wk_max": 1578,
  "cases_ytd": 14123
}
```

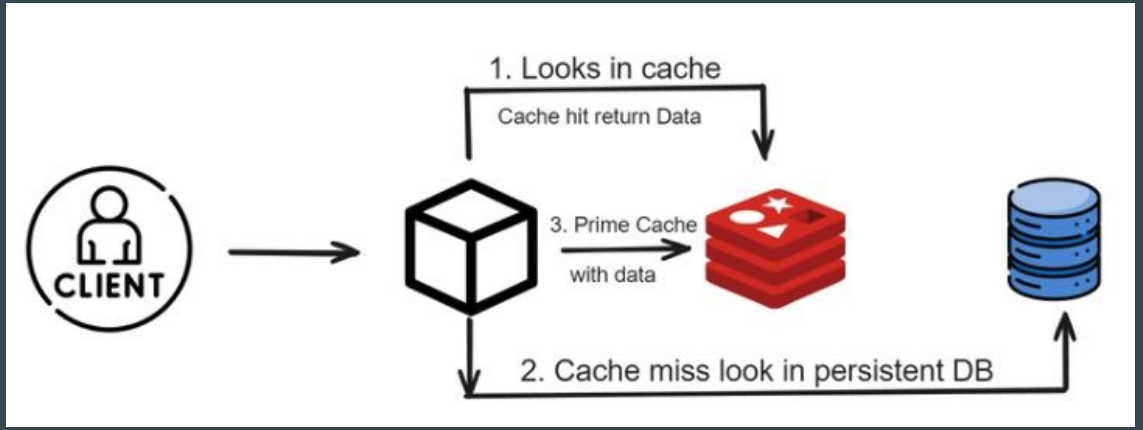
Redis Use Case

Benefits of Redis

- High-Speed Access to Frequently Used Data
- Real-Time Dashboards for Geographic Spread
- Efficient Alerts with Publish/Subscribe

Why not relational?

- Slower speed for real data
- No built-in Pub/Sub feature



Resource: https://dev.to/jemmyasjd/redis-for-beginners-1lnl

Conclusion: Public Health Impact

🎯 **Targeted interventions** : Identify epicenters for resource deployment

🏥 **Clinic planning** : Inform placement of new sexual health services

📊 **Funding prioritization** : Justify resource allocation for education & testing

🔄 **Ongoing monitoring** : Track evolving trends and respond rapidly

Questions?