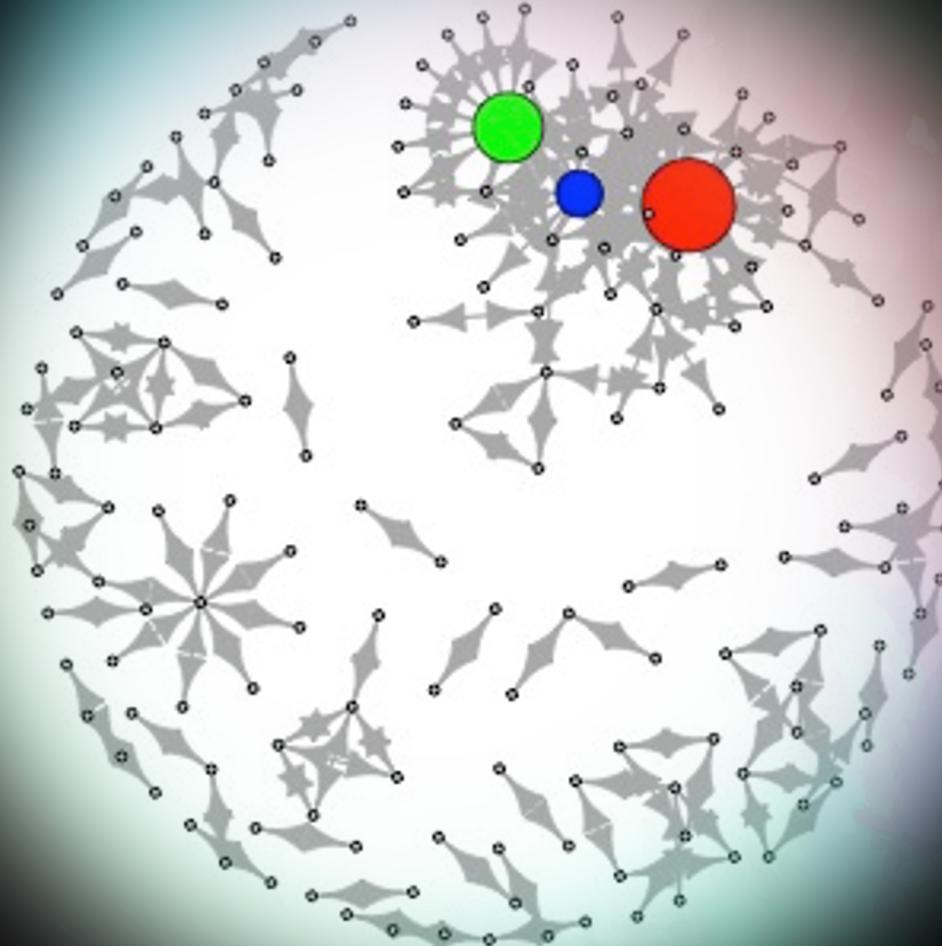


Centrality Analysis of IMDB Dataset



FUNSO OJE & OLUWAFEMI AJEIGBE

IMDB Dataset

7 datasets in repo

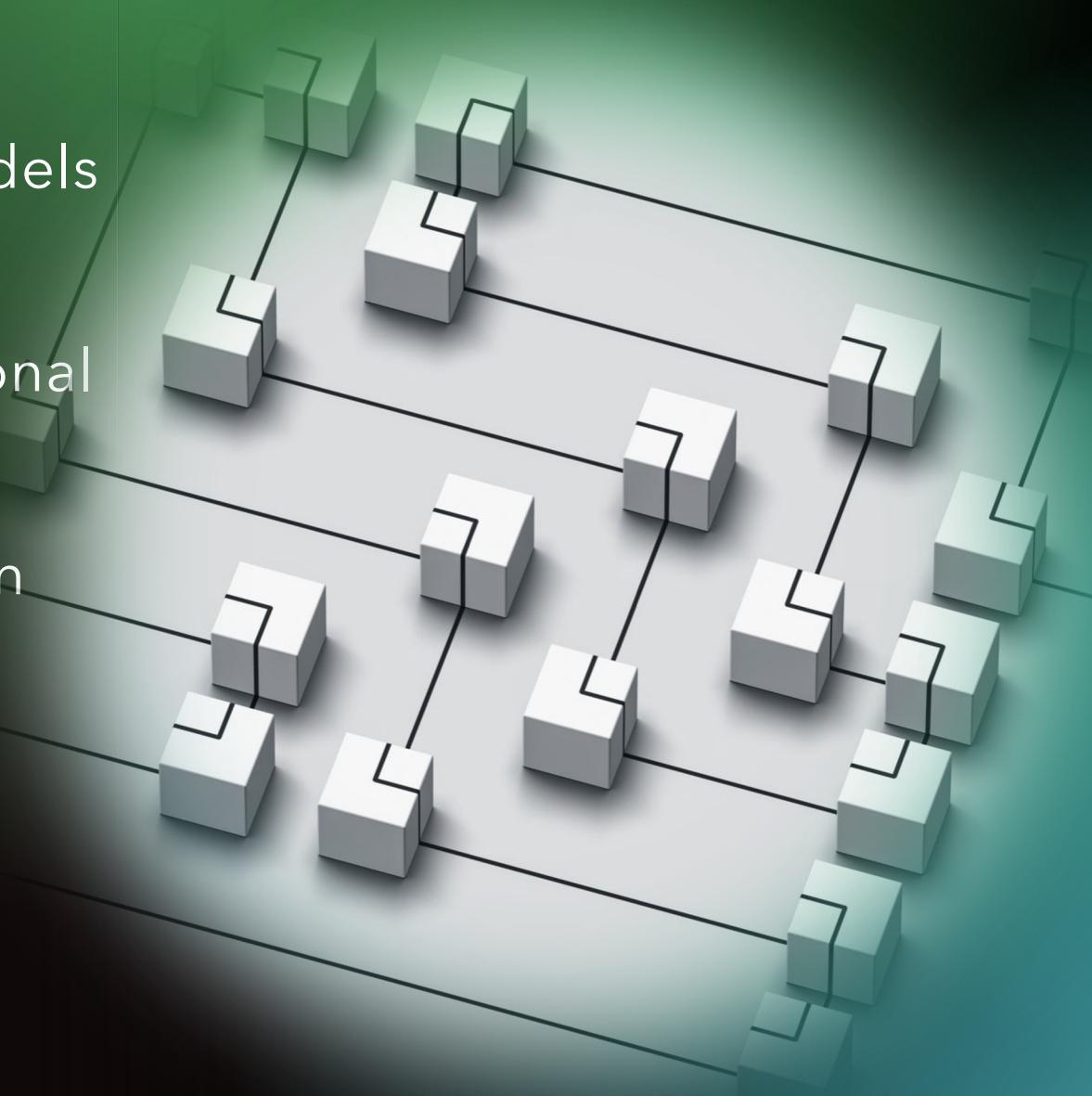
Dataset contains various title info
e.g., rating, actors, crew, year, etc.

Filtered actors and actresses only
(663k+ people)

Filtered movies only (600k+ titles)

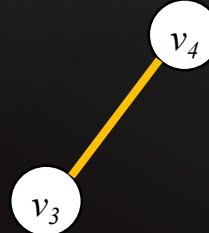
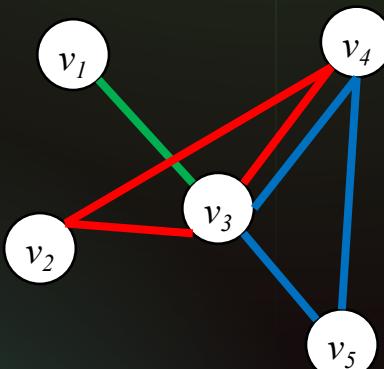
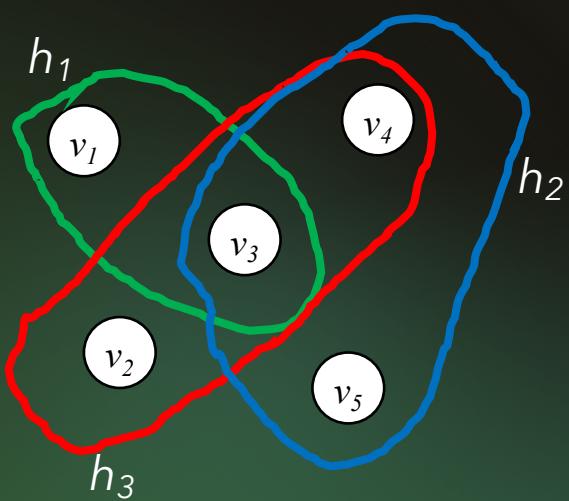
- Dataset Size
- Using Hypergraph - Simple models for Complex Data
- Modelling Complexity in Relational Data
- Using Parallel processing we can achieve faster runtimes

Motivation



Data Analysis

- As a Hypergraph
- Adjacency Edge
- s-line graph ($s=2$)



Brandes BC

- Typically, Betweenness Centrality is $O(n^3)$ time wise and $O(n^2)$ space wise
- Brandes Algorithm runs in $O(nm + n^2\log n)$ time wise and $O(n + m)$ space wise
- Brandes Algorithm explores accumulating dependences rather than dependency summation
- In addition, parallelism is combined with Brandes Algorithm to make the process quicker

```
vector<score_t> bc = nw::graph::exact_brandes_bc<score_t, accu
or<score_t> bc = nw::graph::brandes_bc<decltype(L), score_t, a
vector<score_t> bc = nw::graph::brandes_bc<score_t, accum_t>(L, so
nw::graph::BCVerifier<score_t, accum_t>(L, sources, bc);
std::cout << "Approx. Brande Parallel" << std::endl;
nw::graph::BCVerifier<score_t, accum_t>(L, sources, bc1);
std::cout << "Exact Brande Parallel" << std::endl;
nw::graph::BCVerifier<score_t, accum_t>(L, sources, bc2);
std::cout << "Exact Brande Non Parallel" << std::endl;
nw::graph::BCVerifier<score_t, accum_t>(L, sources, bc3);

();
std::cout << t9 << std::endl;
nat nbc = bc.size();
nat scale = 1.0;
nat nbc
/= ((nbc - 1) * (nbc - 2));
at three scores
at<int> indices(bc.size());
at<int> indices(bc.begin(), indices.end(), 0);
at<int> indices(begin(), indices.begin() + 3, 1);
at<int> indices(begin(), indices.begin() + 3, 1);
```

Results

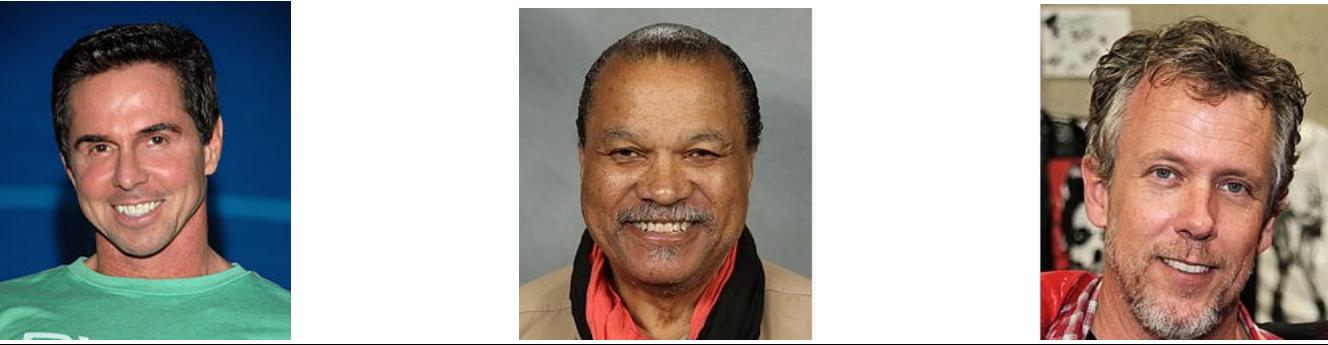


Challenges

- Big Dataset
- $s=1$ issue
- Attempted to use a smaller dataset by filtering by movies made in the US



The US IMDB Dataset



What's next?

01

Validate the result

02

The true centrality result will be when $s=1$

03

Does the s-line approximation help to provide faster centrality results?

04

Where do the actors in a particular s-line fall when $s=1$?

Thank You!

- Questions and Discussions
- Many thanks to Tony and the PNNL team for their support

