Solving Applied Graph Theory Problems in the JuliaGraphs Ecosystem

James P Fairbanks

Oct 24, 2018

Georgia Tech Research Institute

Introduction

The Julia Programming Language

- 1.0 stable release in Aug 2018
- Multiple dispatch
- Dynamic Type system
- JIT Compiler
- Metaprogramming
- Single machine, GPU, and distributed parallelism
- Open Source (MIT License)

Julia Performance Benchmarks

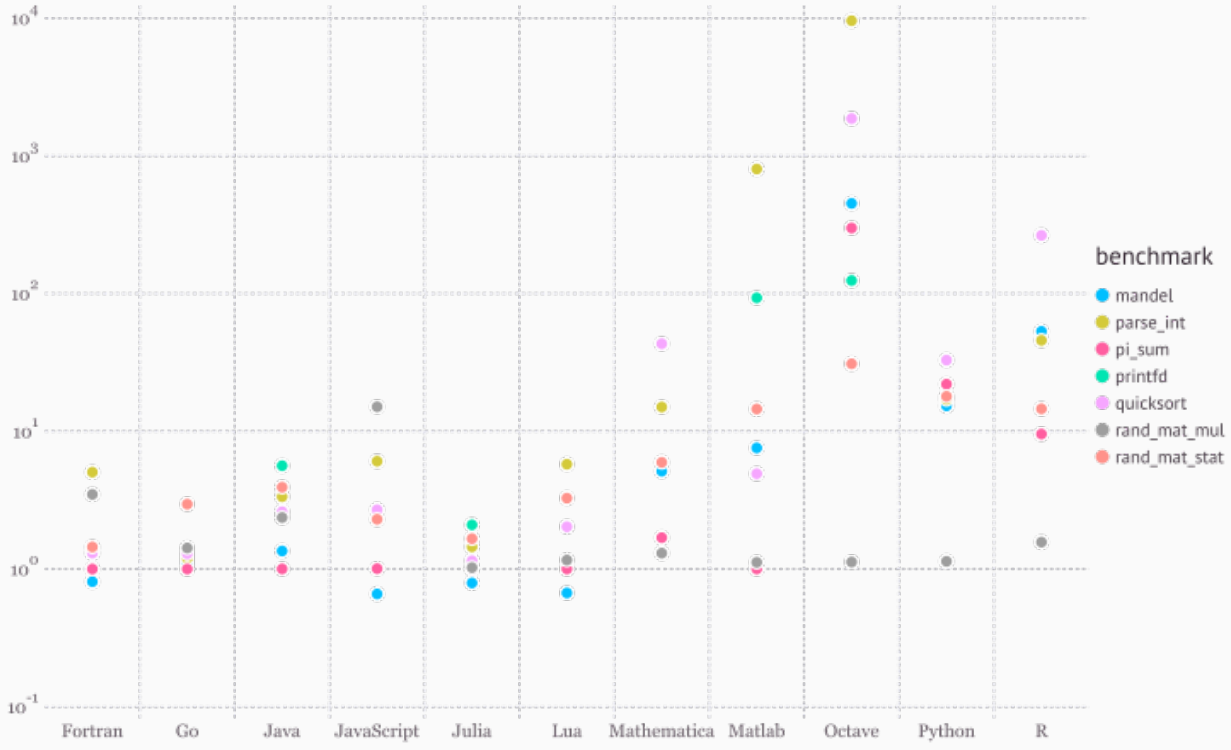


Figure 1: Benchmark times relative to C (smaller is better, C performance = 1.0) (Source: julialang.org)

My path to Julia

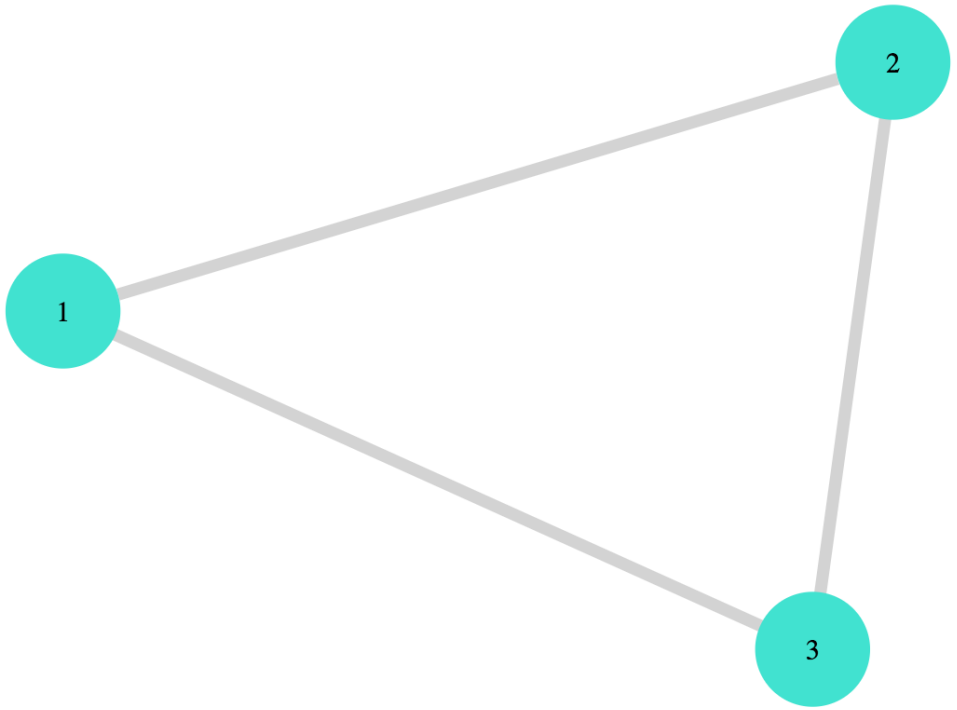
- Started in pure math (chalk),
- Intro programming class (Java)
- Grad school in CSE, Cray XMT (C) C++ was too modern
- Numpy/Pandas bandwagon (Python)
- Numerical Graph Algorithms (Julia)

Outline

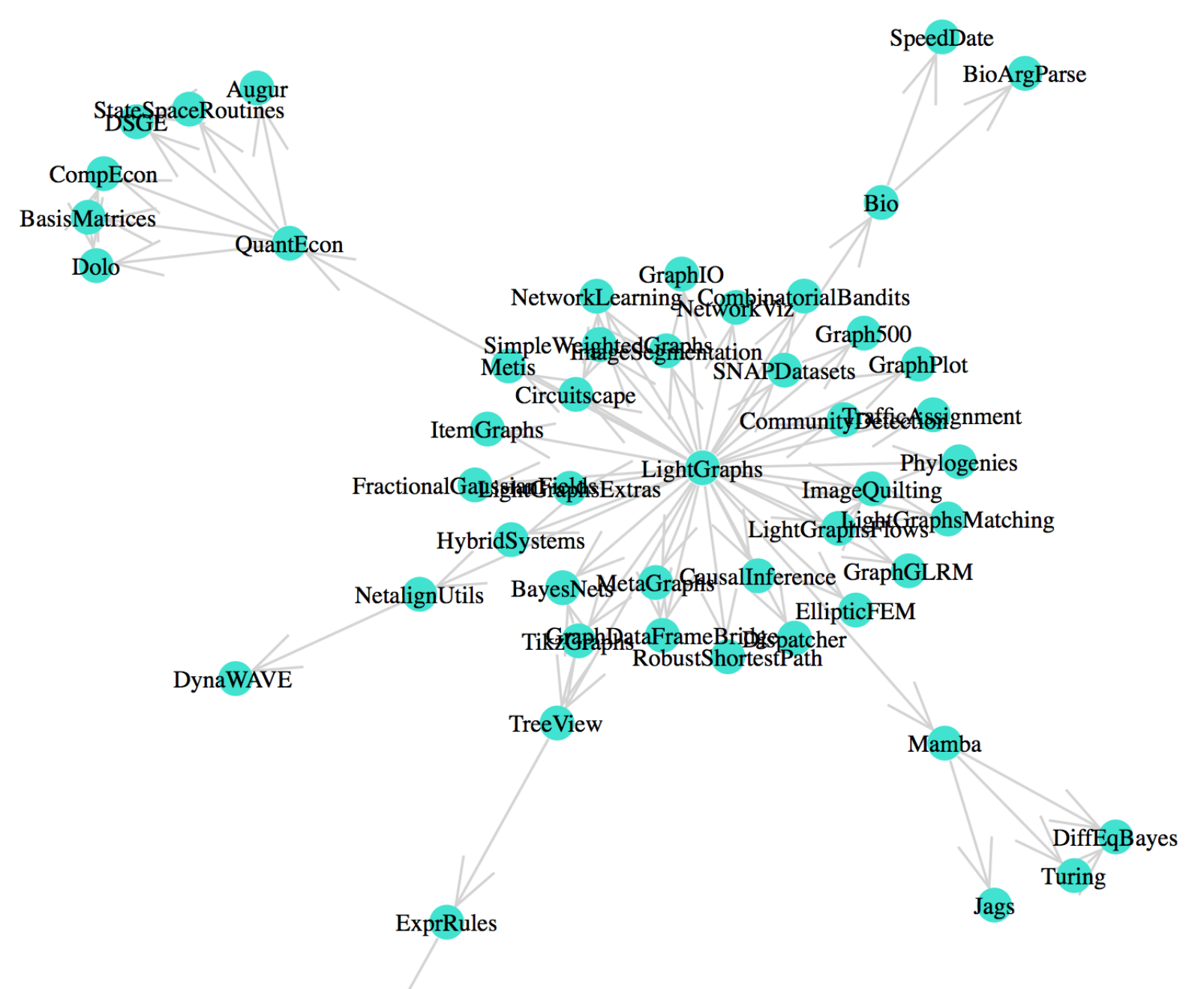
- LightGraphs.jl
- Spectral Clustering
- 2 language problem
- Fake News
- Future Directions

LightGraphs.jl

```julia
using LightGraphs
using GraphPlot

G₁ = Graph(3) # graph with 3 vertices

# make a triangle
add_edge!(G₁, 1, 2)
add_edge!(G₁, 1, 3)
add_edge!(G₁, 2, 3)

gplot(G₁, nodelabel=1:3)
```



LightGraphs.jl is a central vertex



Generic Programing in LightGraphs.jl

Interface for subtypes of AbstractGraph.

- edges
- Base.eltype
- has_edge
- has_vertex
- inneighbors
- ne
- nv
- outneighbors
- vertices
- is_directed

Numerical Analysis for Spectral Partitioning

Spectral Clustering is Graphs + FP

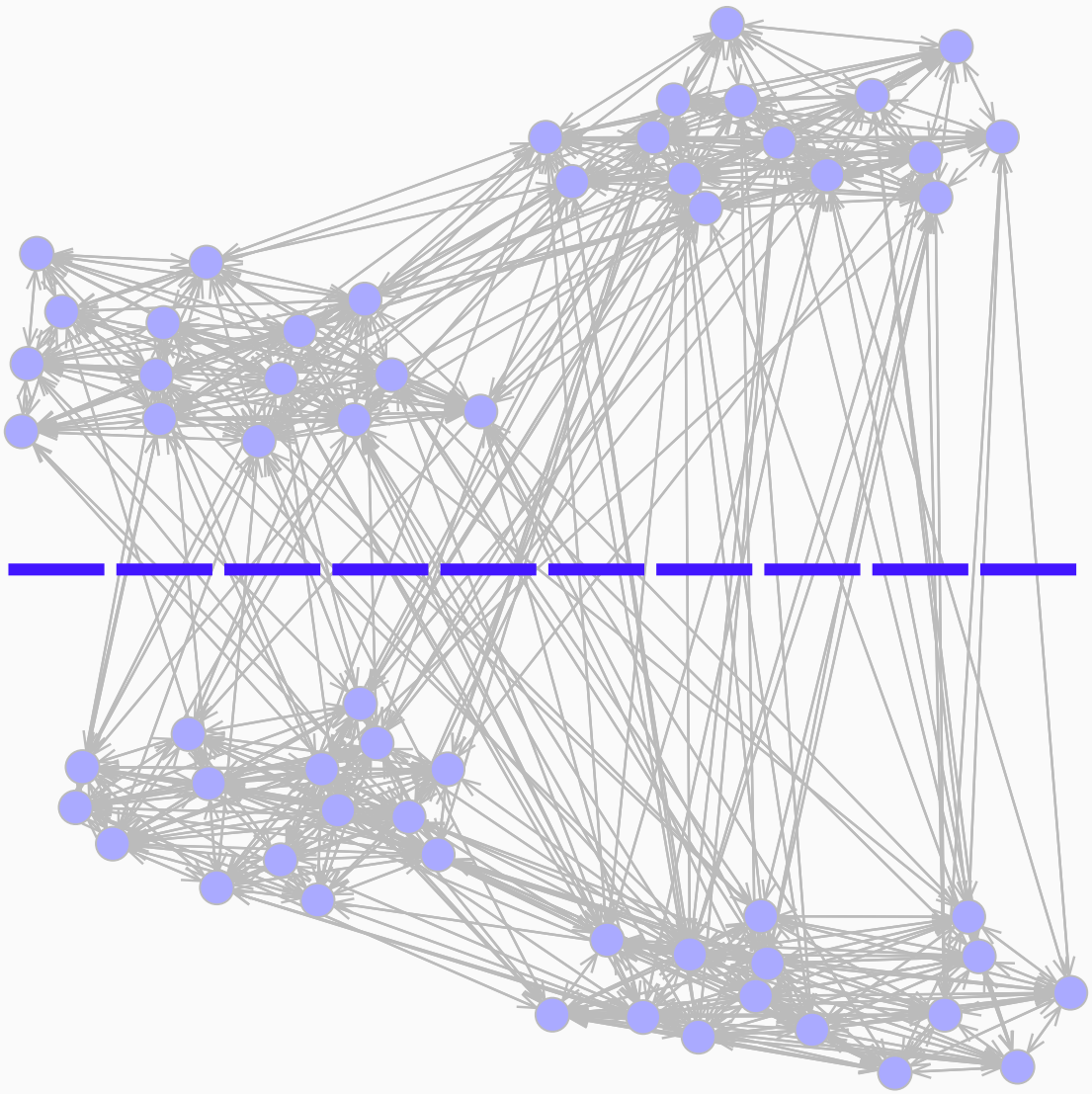


Figure 2: A graph with four natural clusters.

Spectral Graph Matrices

Graphs and Linear Algebra	
Adjacency Matrix	$A_{ij} = \begin{cases} 1, & \text{if } v_i \sim v_j \\ 0, & \text{otherwise} \end{cases}$
Degrees	$D_{ii} = d_i = \deg(v_i)$ all other entries 0.
Combinatorial Laplacian	$L = D - A$
Normalized Laplacian	$\hat{L} = I - D^{-1/2} A D^{-1/2}$
Normalized Adjacency	$\hat{A} = D^{-1/2} A D^{-1/2}$

Spectral Graph Types

```julia
abstract type GraphMatrix{T} end
abstract type Adjacency{T} <: GraphMatrix{T} end
abstract type Laplacian{T} <: GraphMatrix{T} end
struct CombinatorialAdjacency{T,S,V} <: Adjacency{T}
    A::S
    D::V
end
function NormalizedAdjacency(adjmat::CombinatorialAdjacency)
    sf = adjmat.D.^(-1 / 2)
    return NormalizedAdjacency(adjmat, sf)
end
function StochasticAdjacency(adjmat::CombinatorialAdjacency)
    sf = adjmat.D.^(-1)
    return StochasticAdjacency(adjmat, sf)
end
function AveragingAdjacency(adjmat::CombinatorialAdjacency)
    sf = adjmat.D.^(-1)
    return AveragingAdjacency(adjmat, sf)
end
function PunchedAdjacency(adjmat::CombinatorialAdjacency)
    perron = sqrt.(adjmat.D) / norm(sqrt.(adjmat.D))
    return PunchedAdjacency(NormalizedAdjacency(adjmat), perron)
end
```

Spectral Graph Types

```julia
arrayfunctions = (:eltype, :length, :ndims, :size, :strides)
for f in arrayfunctions
    @eval $f(a::GraphMatrix) = $f(a.A)
end
issymmetric(a::GraphMatrix) = issymmetric(a.A)
size(a::GraphMatrix, i::Integer) = size(a.A, i)
issymmetric(::StochasticAdjacency) = false
issymmetric(::AveragingAdjacency) = false

function mul!(Y, adjmat::PunchedAdjacency, x)
    y = adjmat.A * x
    Y[:] = y - dot(adjmat.perron, y) * adjmat.perron
    return Y
end

function mul!(Y, lapl::Laplacian, B)
    mul!(Y, lapl.A, B)
    z = diag(lapl) .* B
    Y[:] = z - Y[:]
    return Y
end
```

Spectral Graph Types

NonBacktracking operator

$$B_{(s,t),(u,v)} = A_{st} * A_{uv} * \delta_{tu} * (1 - \delta_{sv})$$

```julia
struct Nonbacktracking{G <: AbstractGraph}
    g::G
    edgeidmap::Dict{Edge,Int}
    m::Int
end
```

```julia
size(nbt::Nonbacktracking) = (nbt.m, nbt.m)
size(nbt::Nonbacktracking, i::Number) = size(nbt)[i]
eltype(nbt::Nonbacktracking) = Float64
issymmetric(nbt::Nonbacktracking) = false

function *(nbt::Nonbacktracking, x::Vector{T}) where T <: Number
    length(x) == nbt.m || error("dimension mismatch")
    y = zeros(T, length(x))
    for (e, u) in nbt.edgeidmap
        i, j = src(e), dst(e)
        for k in inneighbors(nbt.g, i)
            k == j && continue
            v = nbt.edgeidmap[Edge(k, i)]
            y[v] += x[u]
        end
    end
    return y
end
```

Graph Clustering is Finding Block Structure

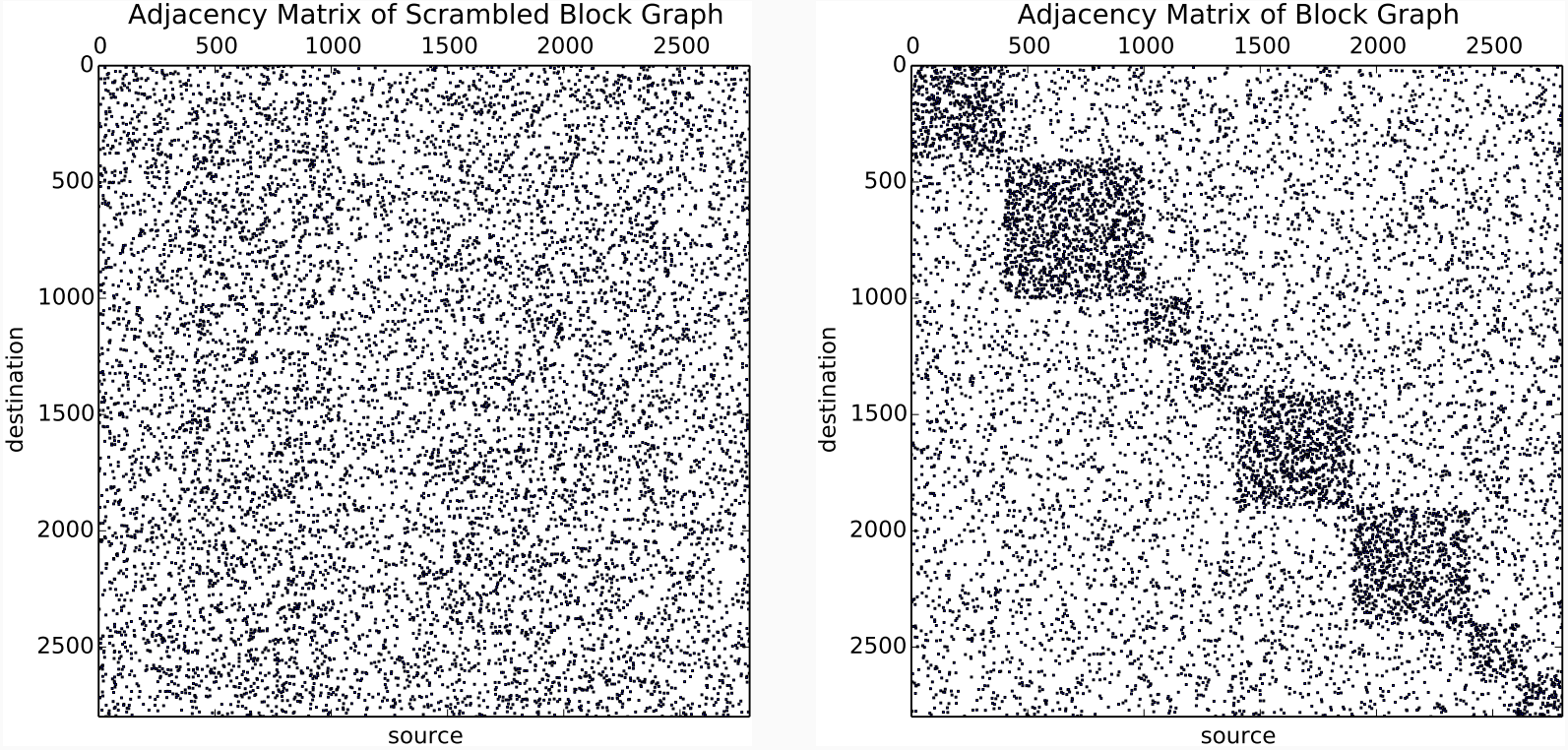


Figure 3: A Stochastic Block Model graph showing the recovery of clusters. On left the adjacency matrix with a random labeling. On right the adjacency matrix permuted to reveal block structure.

The Spectral Sweep Cut Method

- Solve $\hat{L}\mathbf{x} = \lambda_2\mathbf{x}$ to accuracy τ
- Sort $D^{-\frac{1}{2}}\mathbf{x}$
- Check all n sweep cuts of $D^{-\frac{1}{2}}\mathbf{x}$ for best partition

- Recursive bisection applies spectral sweep cuts to each part
- Can achieve any number of clusters
- First step is the bottleneck

- Large τ seems to work in practice
- The goal is to ensure that the final partition recovers the clusters

The Spectral Sweep Cut Method

- Solve $\hat{L}\mathbf{x} = \lambda_2\mathbf{x}$ to accuracy τ
- Sort $D^{-\frac{1}{2}}\mathbf{x}$
- Check all n sweep cuts of $D^{-\frac{1}{2}}\mathbf{x}$ for best partition

- Recursive bisection applies spectral sweep cuts to each part
- Can achieve any number of clusters
- First step is the bottleneck

- Large τ seems to work in practice
- The goal is to ensure that the final partition recovers the clusters

The Spectral Sweep Cut Method

- Solve $\hat{L}\mathbf{x} = \lambda_2 \mathbf{x}$ to accuracy τ
- Sort $D^{-\frac{1}{2}}\mathbf{x}$
- Check all n sweep cuts of $D^{-\frac{1}{2}}\mathbf{x}$ for best partition

- Recursive bisection applies spectral sweep cuts to each part
- Can achieve any number of clusters
- First step is the bottleneck

- Large τ seems to work in practice
- The goal is to ensure that the final partition recovers the clusters

Ring of Cliques

$\mathcal{R}_{b,q}$: q blocks each of size b.

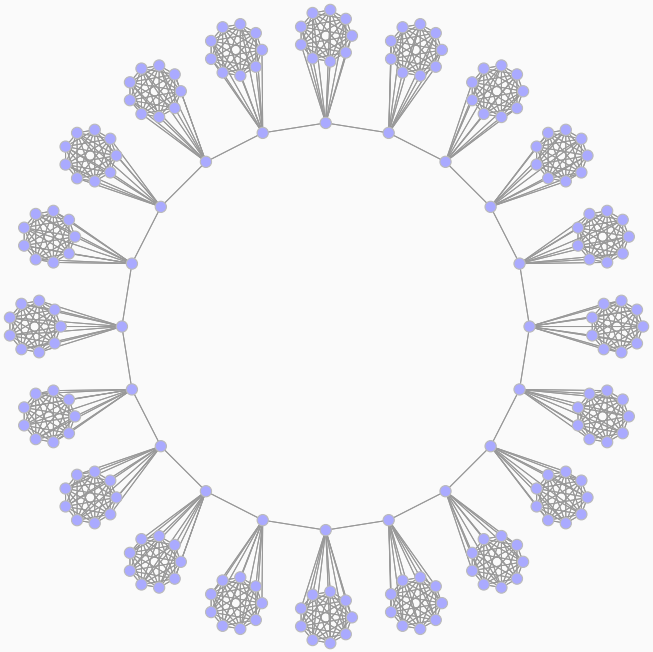


Figure 4: A drawing of $\mathcal{R}_{b,q}$ laid out to show structure.

$$\begin{bmatrix} J_b & \mathbf{e}_1\mathbf{e}_1^t & 0 & \cdots & & \mathbf{e}_1\mathbf{e}_1^t \\ \mathbf{e}_1\mathbf{e}_1^t & J_b & \mathbf{e}_1\mathbf{e}_1^t & 0 & & \\ 0 & \mathbf{e}_1\mathbf{e}_1^t & J_b & \mathbf{e}_1\mathbf{e}_1^t & 0 & \\ \vdots & & \ddots & \ddots & \ddots & \\ & & 0 & \mathbf{e}_1\mathbf{e}_1^t & J_b & \mathbf{e}_1\mathbf{e}_1^t \\ \mathbf{e}_1\mathbf{e}_1^t & & & 0 & \mathbf{e}_1\mathbf{e}_1^t & J_b \end{bmatrix}$$

where $\quad J_b = \mathbf{1}_b\mathbf{1}_b^t - I_b.$

Figure 5: The adjacency matrix of $\mathcal{R}_{b,q}$ has block circulant structure.

Ring of Cliques

$\mathcal{R}_{b,q}$: q blocks each of size b.

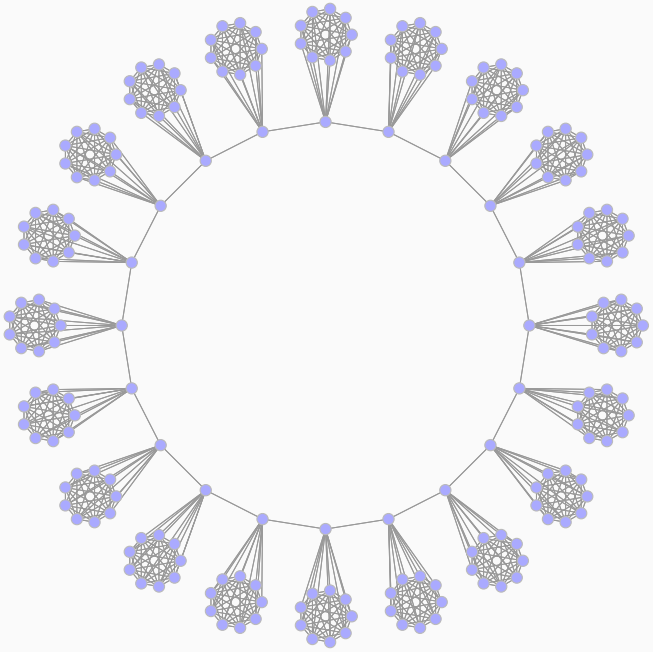


Figure 6: A drawing of $\mathcal{R}_{b,q}$ laid out to show structure.

```julia
struct RingOfBlocks{T} <: AbstractMatrix
    block::T
    q::Int
end

function *(rob::RingOfBlocks, v::AbstractVector)
    z = zeros(eltype(v), size(rob, 1))
    b = size(rob.block, 2)
    for i in 0:rob.q-1
        # Block Diagonal
        z[i*b + (1:b)] = rob.block * v[i*b + (1:b)]
        # Block off diagonal
        z[i*b + 1] += v[(i+1)*b+1]
    end
end

function getindex(rob::RingOfBlocks, i,j)
    b = size(rob.block,2)
    if floor(Int,i/b)*b < j < ceil(Int, i/b)*b
        return rob.block[i%b,j%b]
    end
    if j == floor(Int, i/b) *b - 1
        return 1
    end
    if j == ceil(Int, i/b)*b +1
        return 1
    end
    return 0
end
```

Ring of Cliques Correct Partitions

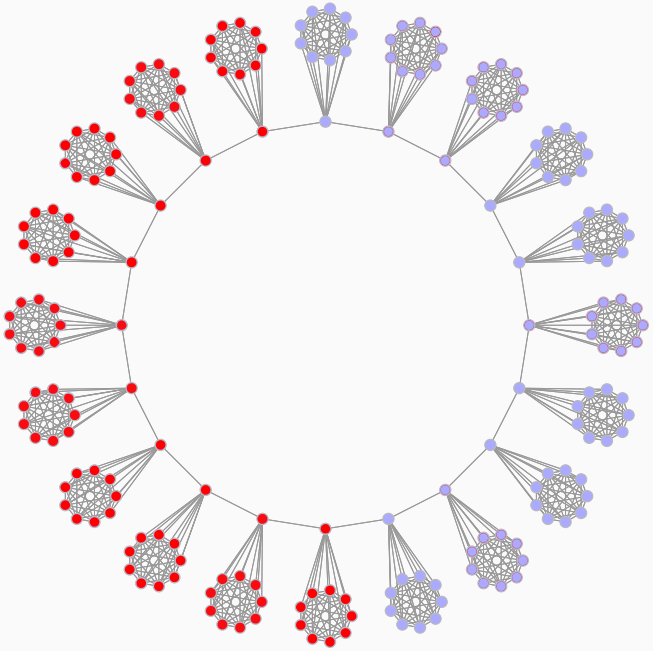


Figure 7: In a "correct" partition of $\mathcal{R}_{b,q}$ all cliques are entirely red or entirely blue.

- $\phi_G = \mathcal{O}\left(b^{-2}q^{-1}\right)$ is optimal
- $\phi_* = \mathcal{O}\left(b^{-2}\right)$ is sufficient to find clusters with recursive bisection sweep cut

Ring of Cliques Correct Partitions

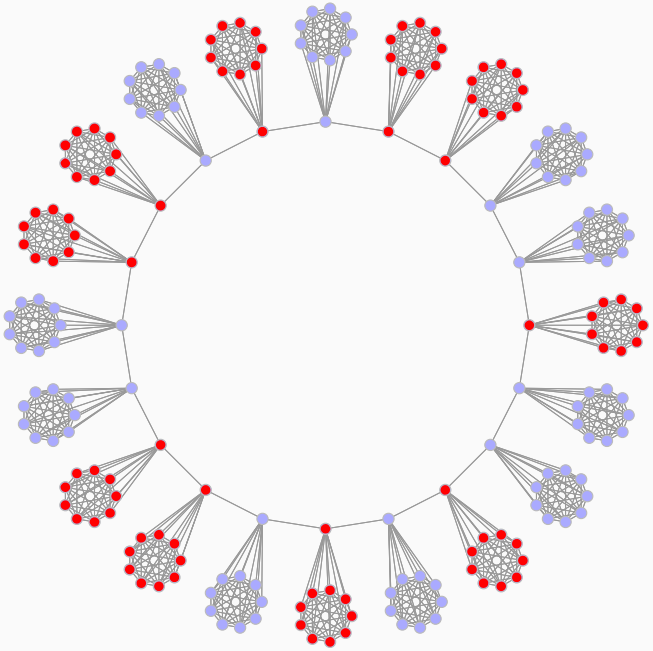


Figure 7: In a "correct" partition of $\mathcal{R}_{b,q}$ all cliques are entirely red or entirely blue.

- $\phi_G = \mathcal{O}\left(b^{-2}q^{-1}\right)$ is optimal
- $\phi_* = \mathcal{O}\left(b^{-2}\right)$ is sufficient to find clusters with recursive bisection sweep cut

Eigenpairs of $\mathcal{R}_{b,q}$ normalized Adjacency Matrix

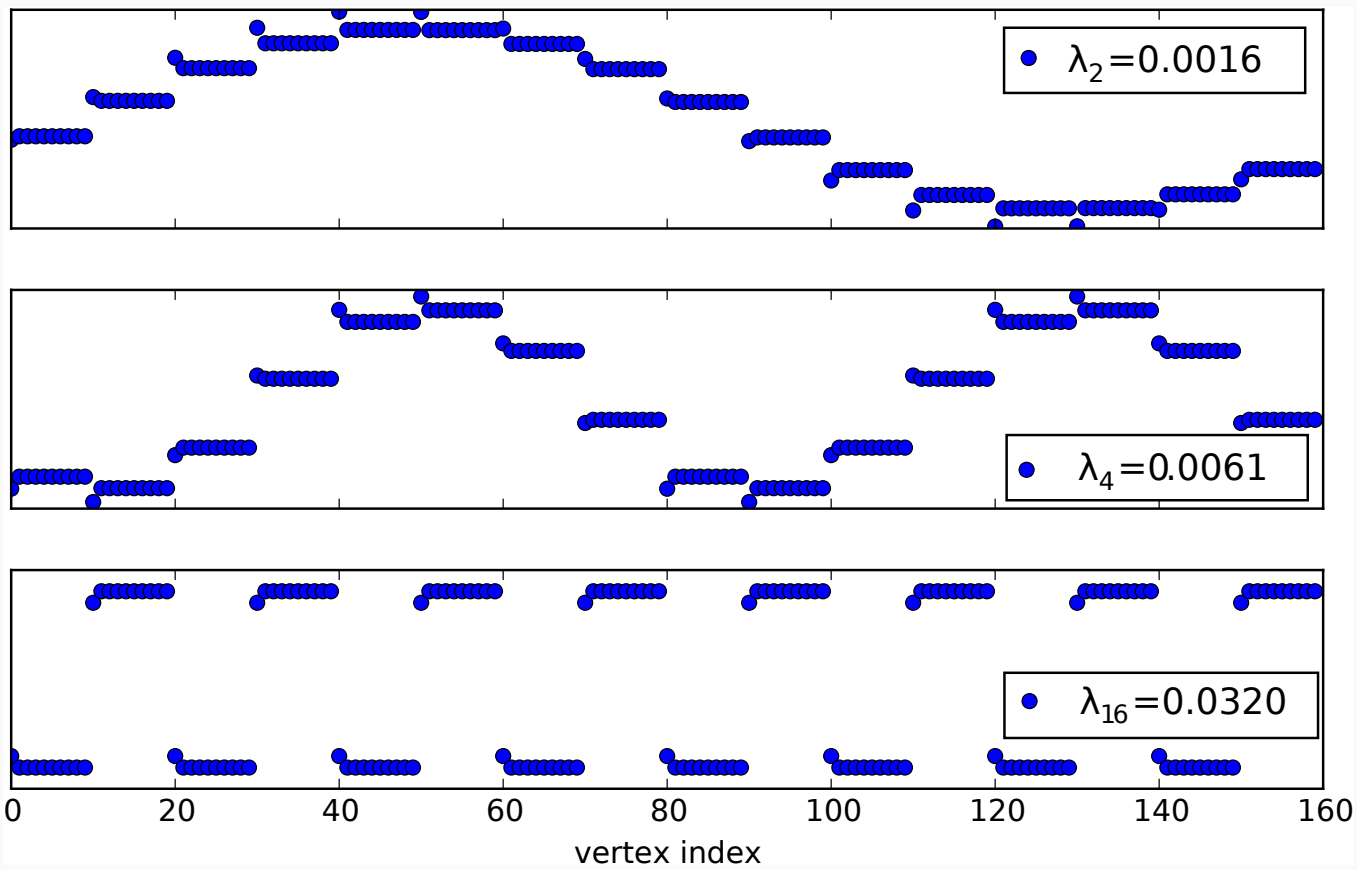


Figure 8: The eigenvectors of $\hat{L}$ for $\mathcal{R}_{10,16}$. The eigenvectors with eigenvalues close to 0 indicate the block structure with differing frequencies.

$\mathcal{R}_{b,q}$ Spectrum

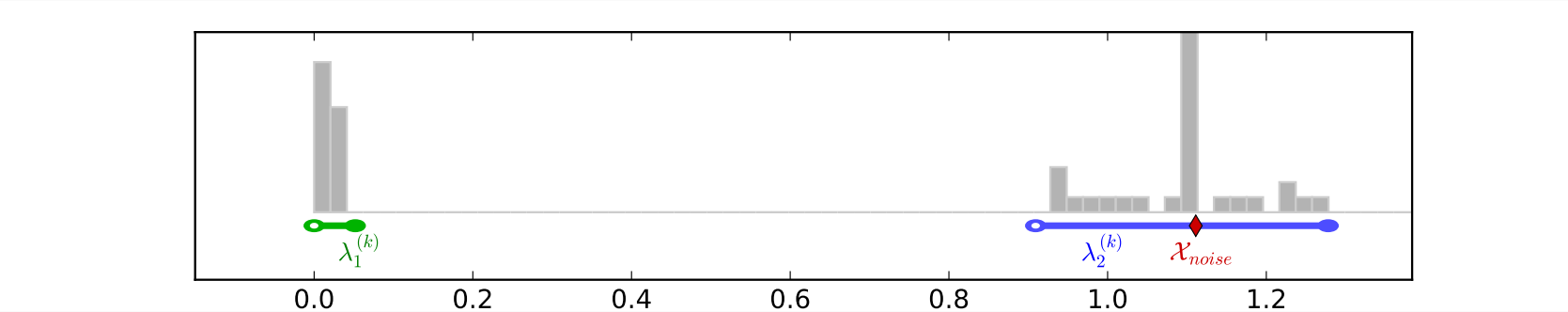


Figure 9: Spectrum of $\hat{L}$ of $\mathcal{R}_{34,10}$.

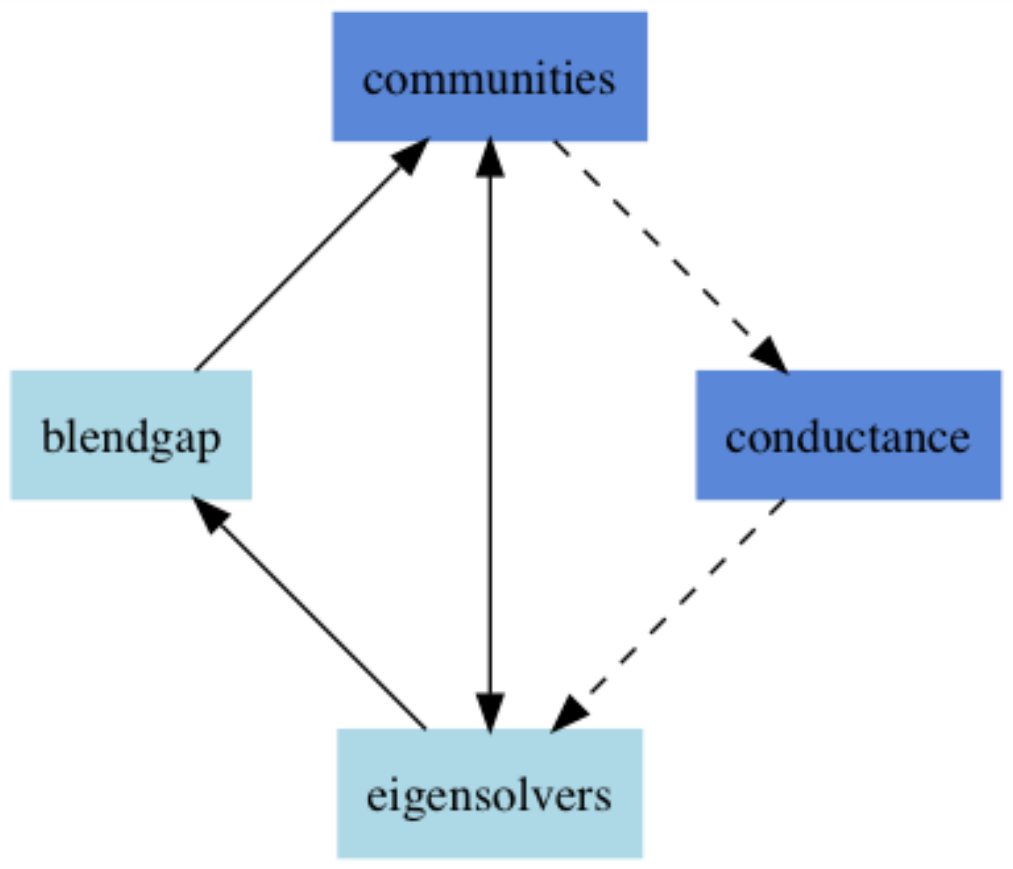
- $[\lambda_2, \lambda_q]$ in green
- $\lambda(\hat{A}) > \frac{1}{2}$ eigenvectors recover partition

b	q	n	δ_2
50	20	1000	1.2275×10^{-4}
500	10	5000	5.1487×10^{-6}

- *Spectral gap* $\delta_2 = \mathcal{O}\left(n^{-2}\right)$
- *Blend gap* $\delta_q = \mathcal{O}\left(1\right)$

Bidirectional learning

We learn about data through algorithms and about algorithms through data.



Legacy Languages Slow Down Research

`ARPACK` is the universally approved eigensolver fortran implementation of `eigs`.

```
c        %-----------------------------------------------%
c        | *  Determine Ritz estimates of the theta.     |
c        |    If RVEC = .true. then compute Ritz estimates |
c        |               of the theta.                   |
c        |    If RVEC = .false. then copy Ritz estimates |
c        |              as computed by dsaupd.           |
c        | *  Determine Ritz estimates of the lambda.    |
c        %-----------------------------------------------%
c
         call dscal (ncv, bnorm2, workl(ihb), 1)
         if (type .eq. 'SHIFTI') then
c
            do 80 k=1, ncv
               workl(ihb+k-1) = abs( workl(ihb+k-1) ) / workl(iw+k-1)**2
 80         continue
c
         else if (type .eq. 'BUCKLE') then
c
            do 90 k=1, ncv
               workl(ihb+k-1) = sigma * abs( workl(ihb+k-1) ) /
     &                          ( workl(iw+k-1)-one )**2
 90         continue
c
         else if (type .eq. 'CAYLEY') then
c
            do 100 k=1, ncv
               workl(ihb+k-1) = abs( workl(ihb+k-1) /
     &                          workl(iw+k-1)*(workl(iw+k-1)-one) )
 100        continue
c
         end if
c
      end if
c
```

`ArnoldiMethod.jl` is a modern Julia implementation

Summary

- Spectral blends lead to lower accuracy requirements in spectral partitioning
- Residual tolerance necessary for Fiedler partitions is $\mathcal{O}\left(n^{-5/2}\right)$ while data clusters are resolved at $\mathcal{O}\left(n^{-1/2}\right)$ for $\mathcal{R}_{b,q}$
- Analyzing model problems for data mining is important for determining accuracy requirements for numerical methods
- Data Analysis structure can be revealed faster than accurate solutions to numerical problem

Novel Stopping Criteria for Spectral Partitioning

Alternative Stopping Criteria

Let $\mathbf{x}^i$ be the i-th iterate and $\mathbf{y}^i = D^{-\frac{1}{2}}\mathbf{x}^i$.

Name	Criterion	Guarantee[1]
Residual	$\left\|\hat{L}\mathbf{x}^i - \mu\mathbf{x}^i\right\| < \tau$	$\sqrt{2\lambda_2 + \tau}$
Difference	$\left\|\phi\left(\mathbf{y}^i\right) - \phi\left(\mathbf{y}^{i+1}\right)\right| < \tau$	—
Naive Cheeger	$\phi\left(\mathbf{y}^i\right) < \sqrt{2\mu}$	trivial
Accurate Cheeger	$\phi\left(\mathbf{y}^i\right) < \sqrt{2(\mu - r)}$	$\sqrt{2\lambda_2}$

Accurate Cheeger reduces iterations by 4X with only an increase in $\phi\left(\mathbf{y}^i\right)$ of less than 2X on average over real world data sets.

[1] Assuming $\mu - \lambda_2 < |\mu - \lambda_i|$ for all $i \neq 2$

Real World Networks Sizes and Eigengaps

Graph Name	V	E	λ_2	λ_3	δ_2
Newman/dolphins	62	318	0.03952	0.23435	0.19483
SNAP/wiki-Vote	8297	103689	0.10055	0.16859	0.06804
Newman/football	115	1226	0.13680	0.18292	0.04612
SNAP/Oregon-2	11806	65460	0.02919	0.04191	0.01272
SNAP/soc-Epinions1	75888	508837	0.00479	0.01323	0.00844
SNAP/cit-HepTh	27770	352807	0.01839	0.02391	0.00552
SNAP/soc-sign-epinions	131828	841372	0.01123	0.01575	0.00452
Newman/lesmis	77	508	0.08813	0.09222	0.00409
SNAP/soc-Slashdot0902	82168	948464	0.01174	0.01321	0.00147
SNAP/email-Enron	36692	367662	0.00353	0.00454	0.00101
Newman/cond-mat-2005	40421	351382	0.00428	0.00484	0.00056
SNAP/web-NotreDame	325729	1497134	0.00134	0.00182	0.00048
SNAP/web-Stanford	281903	2312497	0.00002	0.00008	0.00006
SNAP/web-Google	916428	5105039	0.00043	0.00044	0.00001

Summary

- Stopping when $\phi\left(\mathbf{y}^i\right) \leq \sqrt{2(\mu - r)}$ leads to good performance on real world networks
- Accurate Cheeger reduces iterations by 4X with only an increase in $\phi\left(\mathbf{y}^i\right)$ of less than 2X on average over real world data sets
- Reduction in iterations increases as problems get harder
- Analysis of numerical accuracy and graph analysis objectives leads to better performance
- Learning about algorithms from data and data from algorithms

Julia is an ideal environment

- Hard to write graph algorithms in high level languages
- Hard to inspect and modify numerical algorithm written in low level languages
- Julia lets you solve problem in one language

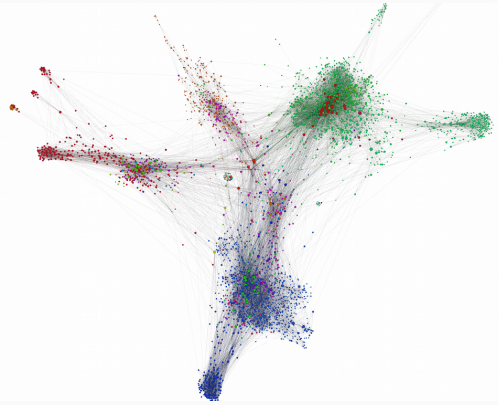
Summary

- Low accuracy eigenvectors are useful for graph partitioning and easier to compute
- Spectral blends are a powerful analytic tool.
- Analyzing numerical accuracy in a graph analysis context leads to improved algorithms and a deeper understanding of computational phenomena

Integrating HPL with HPC Libraries

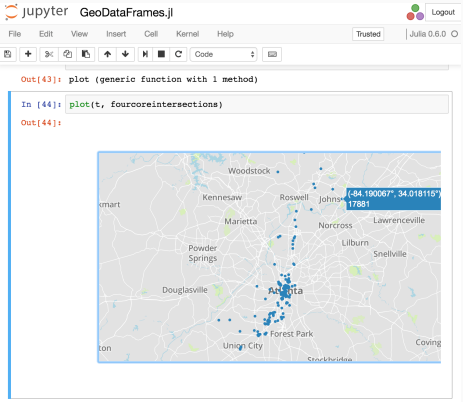
Graph Analysis

- Applications: Cybersecurity, Social Media, Fraud Detection...



(a) Big Graphs

(b) HPC



(c) Productivity

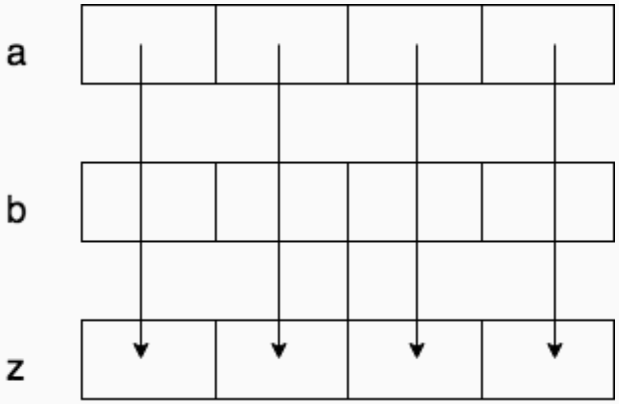
Types of Graph Analysis Libraries

- Purely High productivity Language with simple data structures
- Low level language core with high productivity language interface.

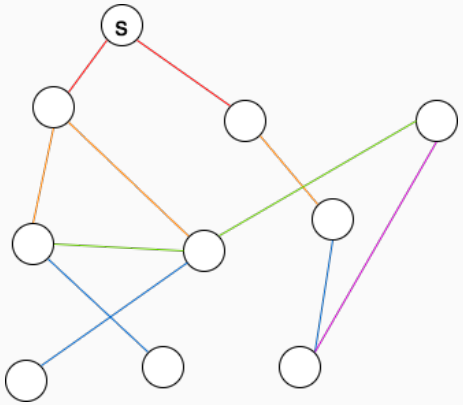
Name	High Level Interface	Low Level Core	Parallelism
SNAP	Python	C++	OpenMP
igraph	Python, R	C	-
graph-tool	Python	C++ (BGL)	OpenMP
NetworKit	Python	C++	OpenMP

Table 1: Libraries using the hybrid model

Why is graph analysis is harder than scientific computing?



(a) $z = exp(a + b^2)$



(b) BFS from s

Figure 11: Computations access patterns in scientific computing and graph analysis

- Less regular computation
- Diverse user defined functions beyond arithmetic
- Temporary allocations kill performance

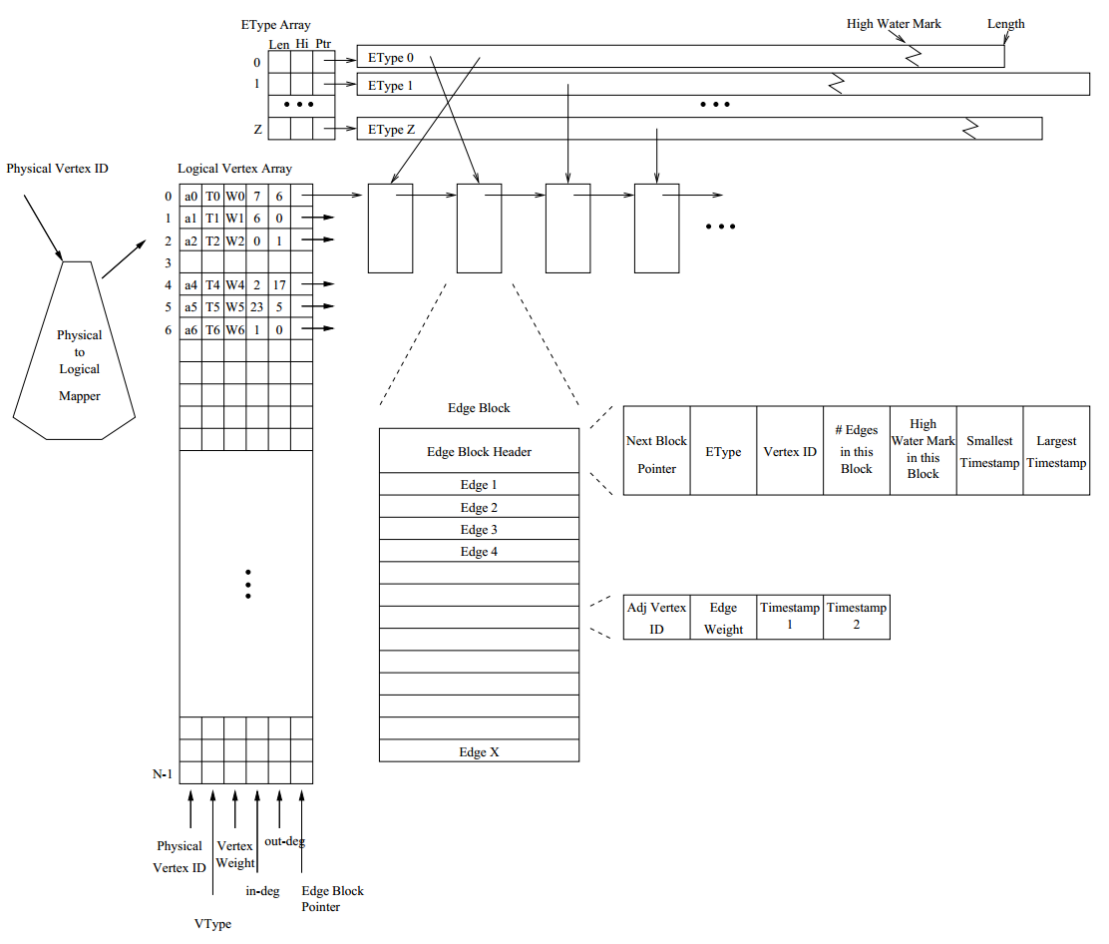
High Productivity Languages

Feature	Python	R	Ruby	Julia
REPL	✓	✓	✓	✓
Dynamic Typing	✓	✓	✓	✓
Compilation	×	×	×	✓
Multithreading	Limited	×	Limited	✓

Table 2: Comparison of features of High Productivity Languages

STINGER

- A complex data structure for graphs in C
- Parallel primitives for graph algorithms



Addressing the 2 language problem using Julia

- Two languages incurs development complexity
- All algorithms in Julia
- Reuse only the complex STINGER data structure from C
- Parallel constructs in Julia, NOT low level languages

Integrating Julia with STINGER

- All algorithms in Julia
- Reuse only the complex STINGER data structure from C
- Parallel constructs in Julia, not low level languages
- Productivity + Performance!

Graph 500 benchmark

- Standard benchmark for large graphs
- BFS on a RMAT graph
  - 2^{scale} vertices
  - $2^{scale} * 16$ edges
- Comparing BFS on graphs from scale 10 to 27 in C and using StingerGraphs.jl
- A multithreaded version of the BFS with up to 64 threads was also run using both libraries

Results Preview

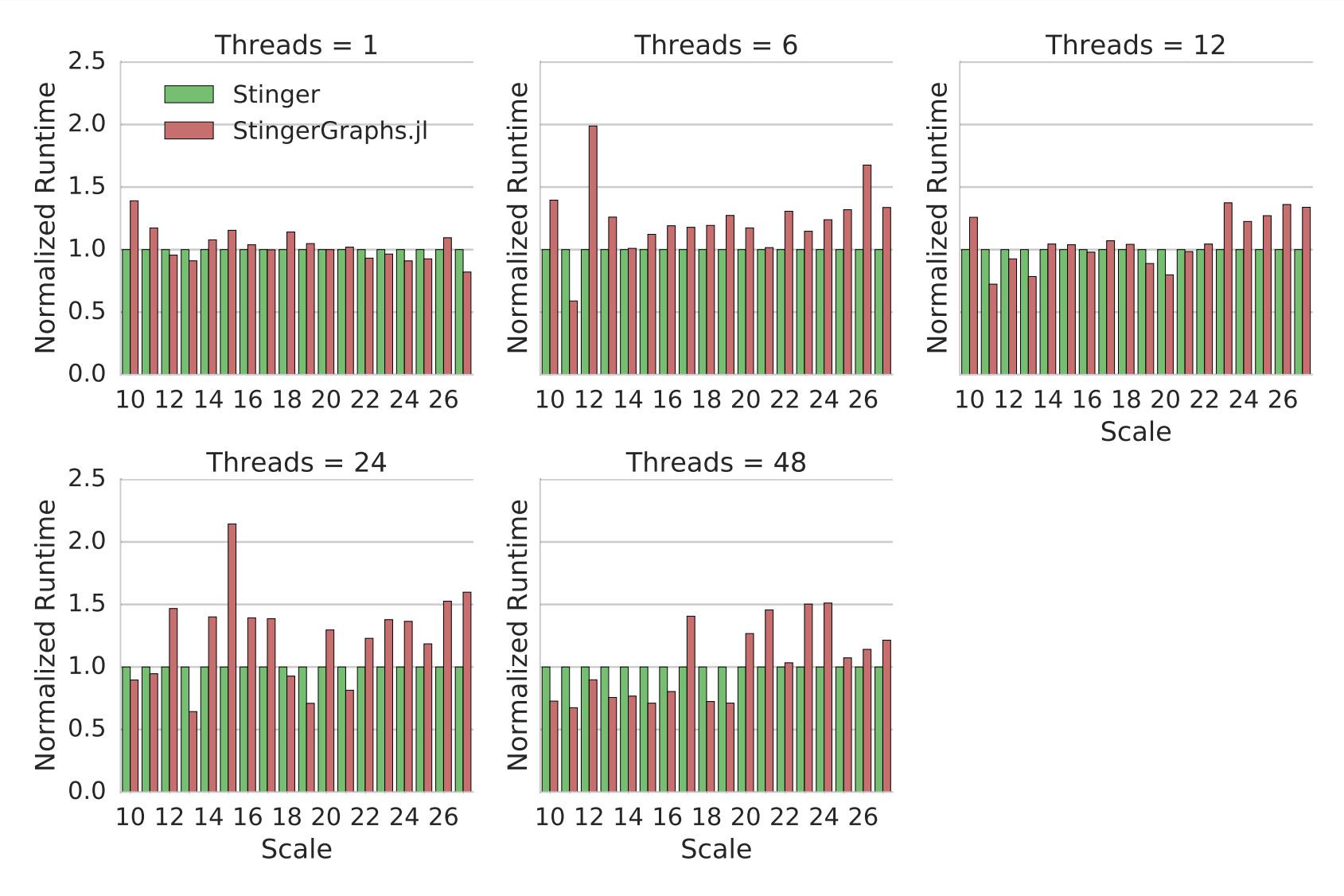


Figure 12: Graph500 Benchmark Results (Normalized to STINGER – C)

Moving data kills performance

Bulk transfer of memory between memory spaces is more expensive than direct iteration

Scale	Exp (I)	Exp (G)	BFS (I)	BFS (G)
10	1.03	2.43	252.17	1833.70
11	2.21	4.92	504.37	3623.40
12	4.64	10.33	1034.36	7239.56
13	9.70	21.04	2142.28	14461.98
14	20.79	44.18	4328.72	28767.98
15	58.11	107.91	12583.00	67962.16
16	127.92	225.55	27036.85	128637.68

Table 3: Iterators (I) vs Gathering successors (G) – all times in ms

Parallelism options in Julia

- **MPI** style remote processes
- **Cilk** style Tasks that are lightweight "green" threads
- **OpenMP** style native multithreading support - `@threads`

We use the `@threads` primitives to avoid communication costs

Julia Atomics

- Atomic type on which atomic ops are dispatched
- `Atomic{T}` contains a reference to a Julia variable of type `T`
- Extra level of indirection for a vector of atomics

```
@eval unsafe_atomic_cas!(x::Ptr{$typ}, cmp::$typ, new::$typ) =
    llvmcall($"""
            %rv = cmpxchg $lt* %0, $lt %1, $lt %2 acq_rel
            ret $lt %rv
            """, $typ, Tuple{Ptr{$typ},$typ,$typ},
            x, cmp, new)
```

Figure 13: Julia provides easy access to LLVM/Clang intrinsics

Unsafe Atomics

Standard atomic types give poor performance, UnsafeAtomics.jl package reduces overhead.

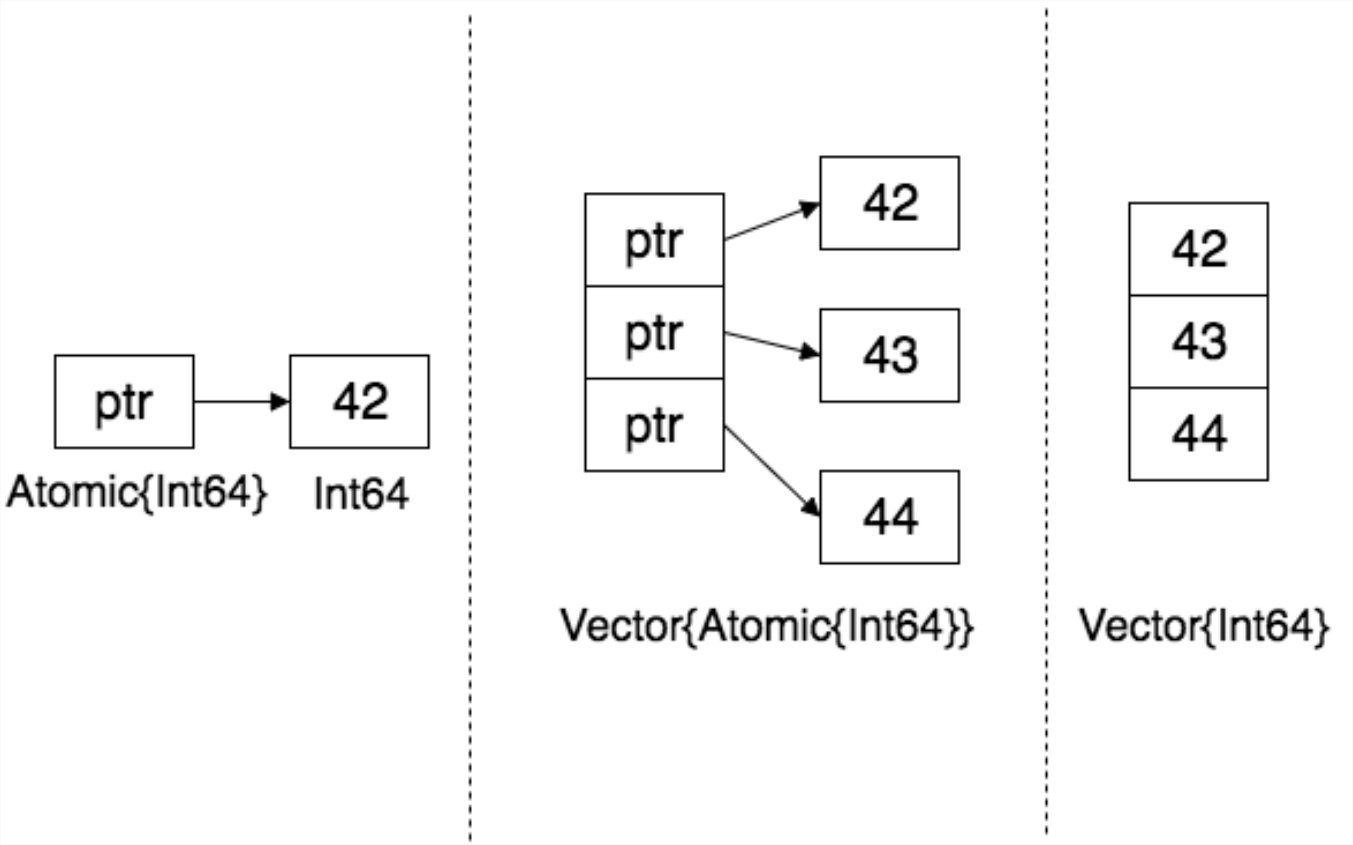


Figure 14: Atomic data structures in Julia

Unsafe Atomics Performance

Scale	Exp (N)	Exp (U)	Exp(N)/ Exp(U)	BFS (N)	BFS (U)	BFS(N)/ BFS(U)
10	0.13	0.1	1.3	47.23	43.27	1.10
11	0.27	0.23	1.17	98.99	91.32	1.08
12	0.62	0.47	1.32	217.44	190.74	1.14
13	1.31	0.97	1.35	505.59	420.84	1.20
14	2.7	2.17	1.24	1158.3	977.1	1.185
15	5.74	3.93	1.46	2576.18	2154.5	1.20
16	11.6	8.77	1.32	5565.87	4559.16	1.22

Table 4: Atomics: Native (N) VS Unsafe (U) (Times in ms)

Results: Parallel run times are comparable for Graph500

Runtimes

Threads	STINGER	Stinger.jl	Slowdown
1	276.46	250.18	0.90x
6	169.93	237.21	1.40x
12	140.53	185.74	1.32x
24	97.73	145.83	1.49x
48	86.41	103.08	1.19x

Table 5: Total time to run Graph500 BFS benchmark for all graphs scale 10-27, in minutes

Results: Parallel Scaling is competitive with OpenMP

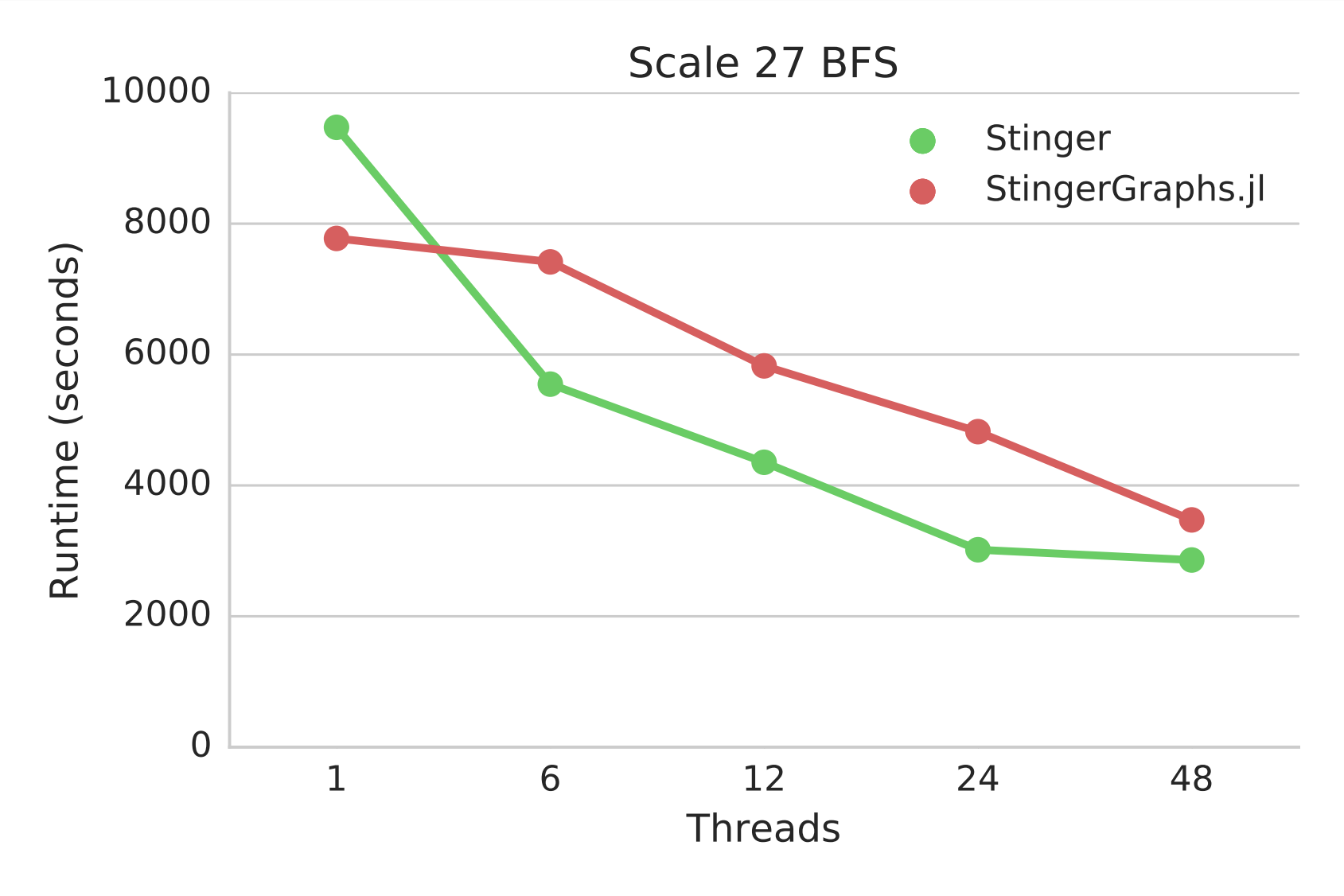


Figure 15: Performance scaling with threads

Summary

- Tight integration between high productivity and high performance languages is possible
- Julia is ready for HPC graph workloads
- Julia parallelism can compete with OpenMP parallelism
- We can expand HPC in High Level Languages beyond traditional scientific applications

Fast Abstractions Open New Possibilities

- Stinger has a huge problem: inflexibility
- Because it was written in C for the XMT with no abstraction
- Can't change the types of the edge weights

Stinger has no Abstraction

```julia
struct StingerEdge
    neighbor::Int64
    weight::Int64
    timefirst::Int64
    timerecent::Int64
end

struct StingerEdgeBlock
    next::UInt64
    etype::Int64
    vertexid::Int64
    numedges::Int64
    high::Int64
    smallstamp::Int64
    largestamp::Int64
    cache_pad::Int64
end
```

- Stinger metadata is totally rigid
- MetaGraphs.jl uses dictionaries for each edge like NetworkX

Finding Fake News with Math

Fake News Flavors

The New York Times.

NEW YORK, MONDAY, OCTOBER 31, 1938.

MEAD STANDS PAT AS A NEW DEALER IN BID FOR SENATE

Democratic Candidate Opposes Any Except Minor Changes in Labor and Security Laws

UPHOLDS THEORY OF TVA

Radio Listeners in Panic, Taking War Drama as Fact

Many Flee Homes to Escape 'Gas Raid From Mars'—Phone Calls Swamp Police at Broadcast of Wells Fantasy

OUSTED JEWS FIND REFUGE IN POLAND AFTER BORDER STAY

Exiles Go to Relatives' Homes or to Camps Maintained by Distribution Committee

REVEAL CRUELTY OF TRIP

ORIGINAL!

Facebook To Ban Fake News, Says Mark Zuckerberg While Eating Dolphin

Facebook will take measures to remove fake news stories from peoples' news feeds, Mark Zuckerberg announced this morning while consuming a live dolphin.

In 2016, the prevalence of political fact abuse – promulgated by the words of two polarizing presidential candidates and their passionate supporters – gave rise to a spreading of fake news with unprecedented impunity.

Fake news: Hillary Clinton is running a child sex ring out of a pizza shop.

Fake news: Democrats want to impose Islamic law in Florida.

Fake news: Thousands of people at a Donald Trump rally in Manhattan chanted, "We hate Muslims, we hate blacks, we want our great country back."

None of those stories – and there are so many more like them – is remotely true.

Fake News and the Modern Web

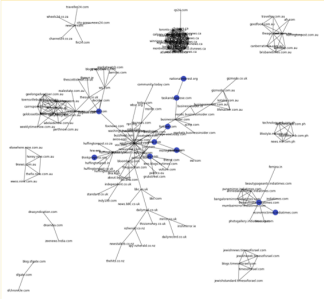
- MOTIVE: Clickbait revenue streams and political campaign funding incentivizes low quality articles to attract readers
- MEANS: The democratization of online media allows anyone to setup a website and publish unadjudicated content
- OPPORTUNITY: Social media has decentralized the enforcement of ethical norms in journalism

Manual fact-checking is valuable but suffers from slow reaction times and accusations of bias.

Approach

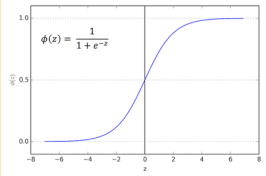
"Structural Method"

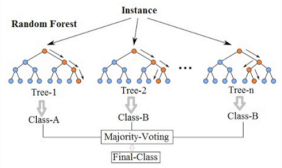
Probabilistic inference using a domain (publisher) web link network

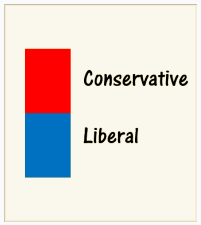


"Content Model"

Traditional supervised learning classifiers using textual features

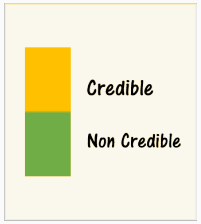






"Bias Detection"

Class label denotes political ideology

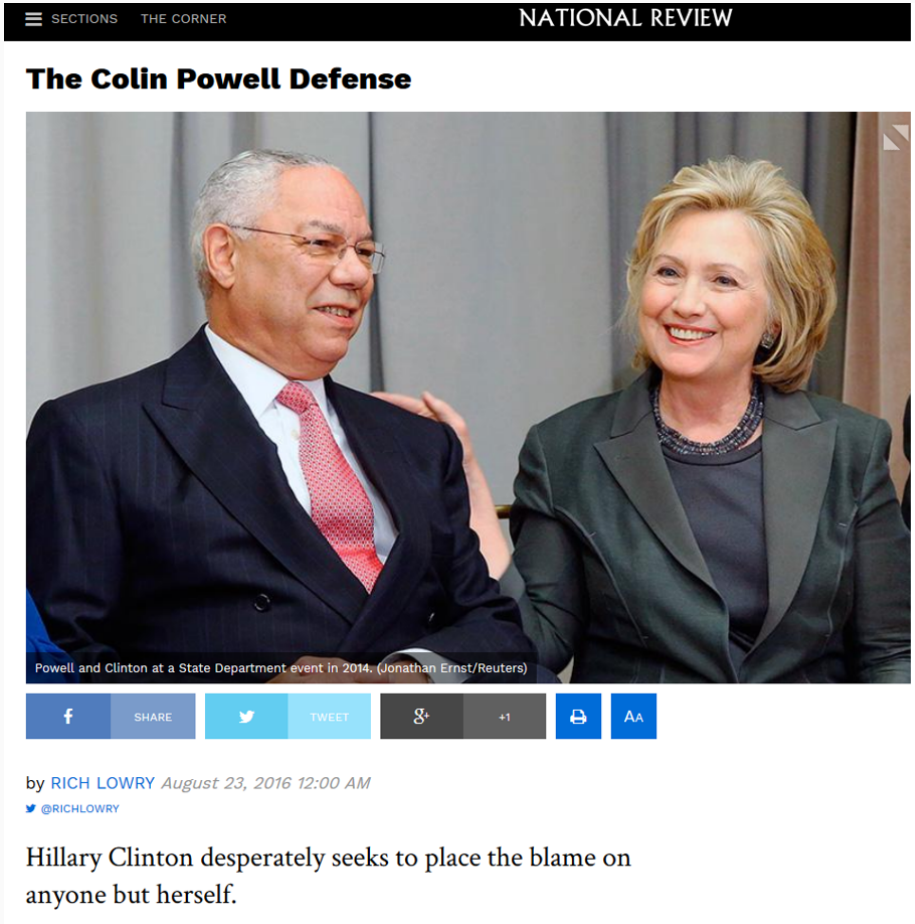


"Credibility Assessment"

Class label denotes source reputation

Bias Detection

Humans can pick up on nuanced but powerful signals of bias in terms of *semantics*, *sentiment (tone)* and *content*.



SECTIONS THE CORNER NATIONAL REVIEW

The Colin Powell Defense

Powell and Clinton at a State Department event in 2014. (Jonathan Ernst/Reuters)

SHARE TWEET +1 AA

by RICH LOWRY *August 23, 2016 12:00 AM*

@RICHLOWRY

Hillary Clinton desperately seeks to place the blame on anyone but herself.

WAR STORIES MILITARY ANALYSIS. JULY 6 2016 1:44 PM

The Hillary Clinton Email Scandal Was Totally Overblown

SLATE

We learned nothing new from the investigation or James Comey's statement.

By Fred Kaplan

Hillary Clinton in New York on March 9, 2015.

Lucas Jackson/Reuters

Content Model

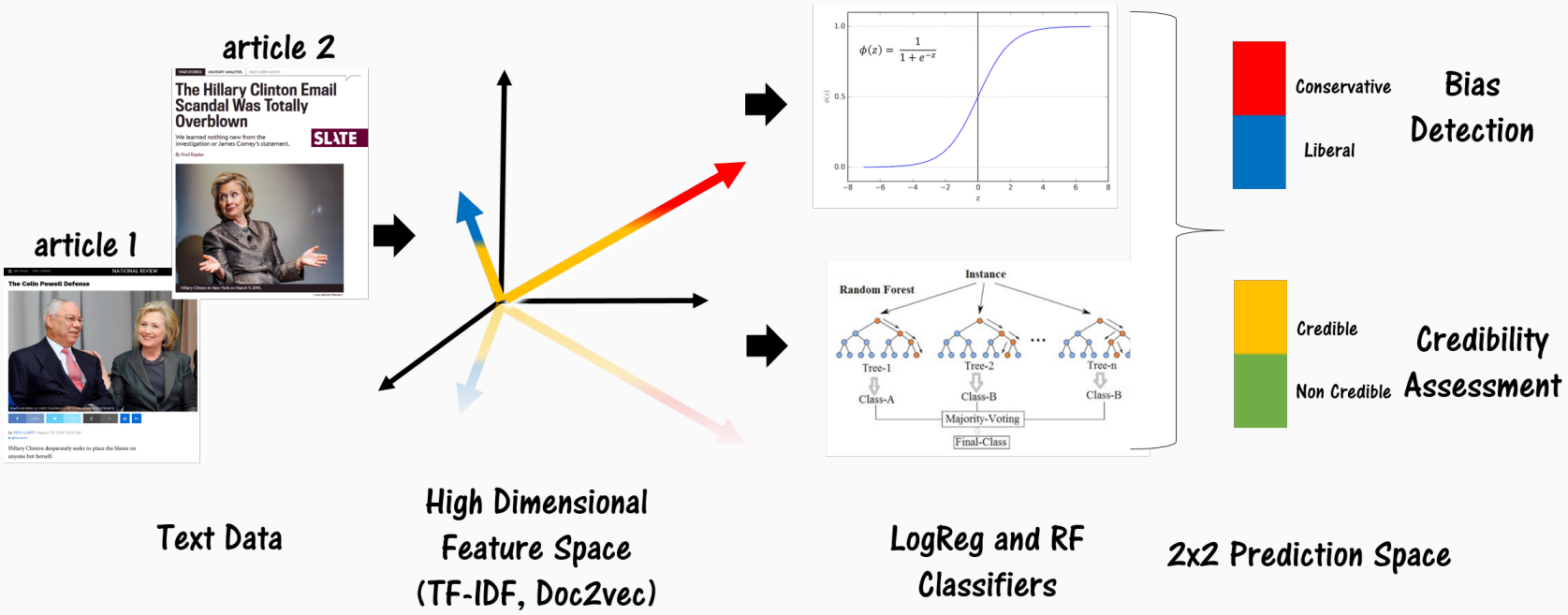


Figure 16: Content Model Pipeline

Credibility Assessment: Fake or Not Fake?

Oprah Stokes 2020 Rumors with Tweet: 'Thanks for Your Vote'

September 29, 2017

SHARE TWEET

Could the United States go from a reality TV president to a daytime TV president? If Oprah Winfrey's recent Twitter activity is any indication, it's at least a possibility.

The former queen of daytime television ignited rumors of a presidential bid after tweeting out a New York Post article that calls her the "Democrats' best hope" to beat President Donald Trump in 2020.

> *"Thanks for your VOTE of confidence!" Winfrey added, tagging the author of the piece:*

Oprah Winfrey
@Oprah

@jpodhoretz Thanks for your VOTE of confidence!Democrats' best hope for 2020: Oprah | New York Post
nypost.com/2017/09/27/dem…
6:49 PM - Sep 28, 2017

Democrats' best hope for 2020: Oprah
On Sunday night's "60 Minutes," a panel of Michigan voters spent 20 minutes discussing their political differences on screen. It was a moving
nypost.com

802 1,401 4,260

When words are not enough...

Source: HillaryDaily.com

Oprah Stokes 2020 Rumors with Tweet: 'Thanks for Your Vote'

September 29, 2017

SHARE TWEET

Could the United States go from a reality TV president to a daytime TV president? If Oprah Winfrey's recent Twitter activity is any indication, it's at least a possibility.

The former queen of daytime television ignited rumors of a presidential bid after tweeting out a New York Post article that calls her the "Democrats' best hope" to beat President Donald Trump in 2020.

> *"Thanks for your VOTE of confidence!" Winfrey added, tagging the author of the piece:*

Oprah Winfrey
@Oprah

@jpodhoretz Thanks for your VOTE of confidence!Democrats' best hope for 2020: Oprah | New York Post
nypost.com/2017/09/27/dem...
6:49 PM - Sep 28, 2017

Democrats' best hope for 2020: Oprah
On Sunday night's "60 Minutes," a panel of Michigan voters spent 20 minutes discussing their political differences on screen. It was a moving
nypost.com

802 1,401 4,260

- Trump Slams 'Crooked Hillary' for Making Excuses for Losing the Election
- New Book Reveals That Obama Pushed Hillary to Concede in 2016 Election
- 2016 Democratic Presidential Candidate Blasts Media for Being Against Trump 'Right from the Beginning'
- NYT Report Suggests Trump & Russia Conspired During Election–But Then There's the Fine Print...
- Michelle Obama: If I Ran Against Trump I Would Have Beaten Him Easily!
- Chelsea's Husband Shutters Hedge Fund

Structural Analysis Example: Stars

- Sites of degree 1 with one central, common site
- Commonly seen in CSS link networks

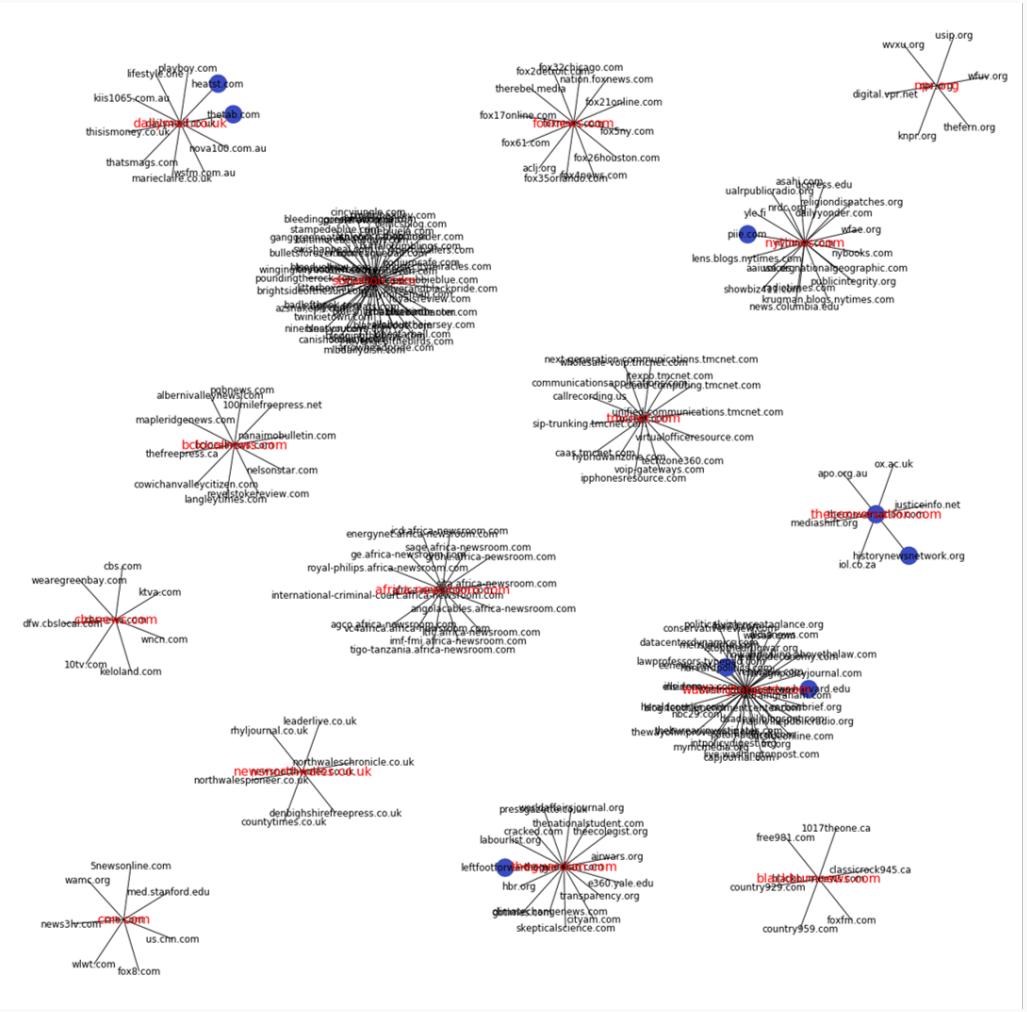


Figure 17: Shared CSS sourced from Big News Network

Structural Method

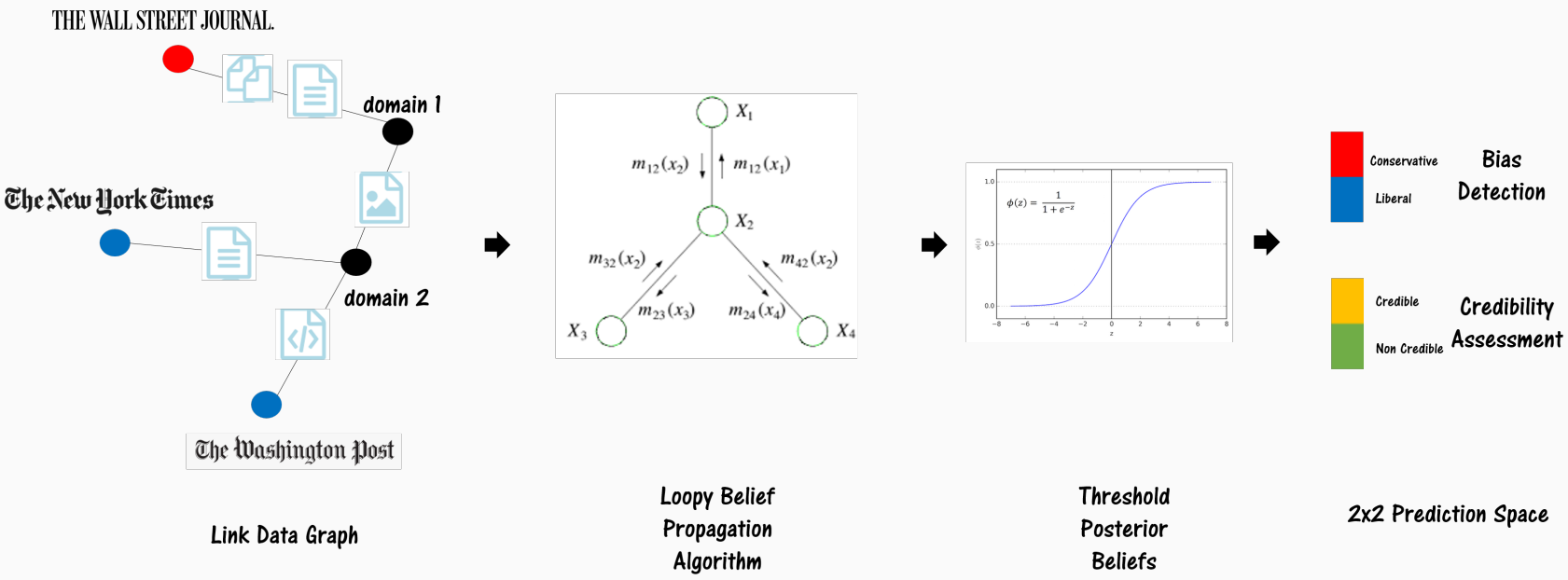


Figure 18: Structural Method Pipeline

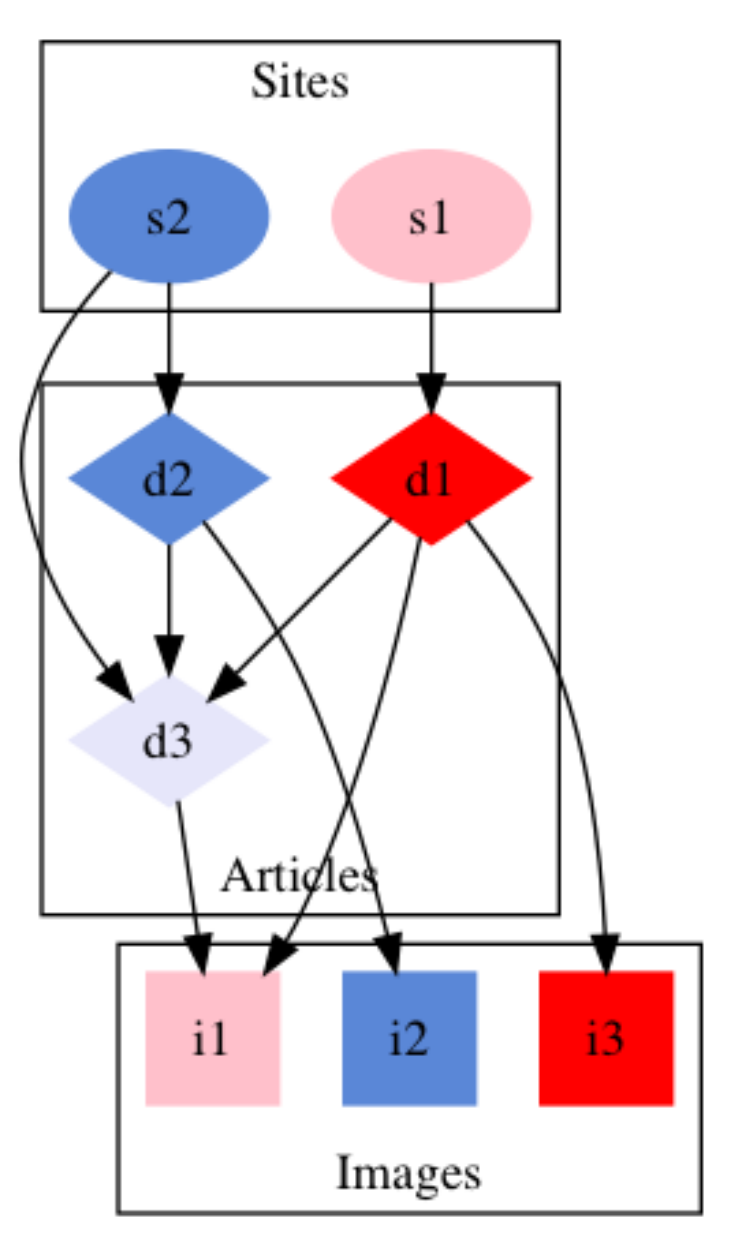
Structural Method: Graph Creation

HTML Tag	Description
`<a>`	Mutually linked sites (text content)
`<link>`	Shared CSS (visual style)
`<script>`	Shared JavaScript files (user interaction)
`<img>`	Common images, logos, or icons (visual content)

Table 6: Link Types used in Graph Construction

- An undirected and unweighted graph was constructed using link structure from 19,786 domains (nodes) with 32,632 links (edges)
- Social media sites and buttons as well as sites that only linked back to themselves were excluded

Structural Method: Belief Propagation




```julia
struct BeliefProblem{T<:Real, U<:Integer, G<:AbstractGraph}
    m::AbstractArray{T,3}
    ϕ::AbstractArray{T,2}
    ψ::AbstractArray{T,2}
    A::G
    X::AbstractArray{U, 1}
end
```

- Belief Propagation is tensor math defined in terms of a graph
- Ideal use for Julia because libraries are easy to combine

Structural Method: Belief Propagation

BP is run separately for each classification task and relies on the following:

- Node potential function: $\phi(x_i)$ "*a priori* node i's class assignment to class x_i"
- Edge potential function: $\psi_{ij}(x_i, x_j)$ "probability node j belongs to class x_j given node i belongs to class x_i"

$\psi_{ij}(x_i, x_j)$	x_i	x_j
x_i	1-ϵ	ϵ
x_j	ϵ	1-ϵ

- Nodes pass messages: $m_{ij}(x_j)$ "node i's belief about node j belonging to class x_i"

$$m_{ij}(x_j) \leftarrow \sum_{x_i \epsilon X} \phi(x_i)\psi_{ij}(x_i, x_j) \prod_{k \epsilon N(i)/j} m_{ki}(x_i) \qquad (1)$$

Experiments: Media Bias Fact Check

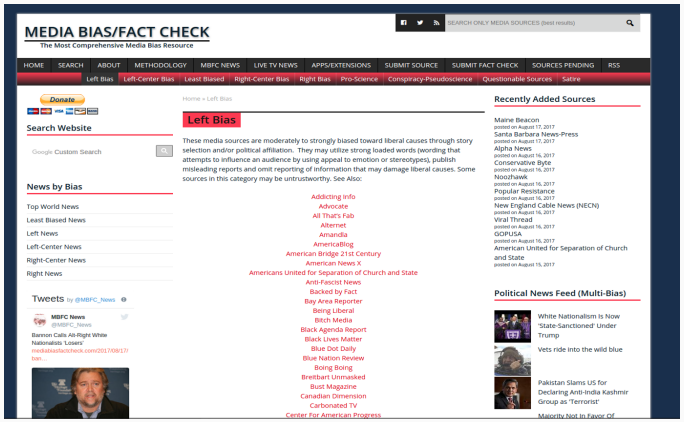


Figure 19: Volunteer run fact checking site `mediabiasfactcheck.com`

- Rubric used to produce ratings for domains for 4 categories:
  - Biased wording/headlines
  - Factual/Sourcing
  - Story Choices (e.g. inclusive of multiple perspectives)
  - Political Affiliation/Endorsement
- Available labels were scraped and converted to binary labels for our classification tasks

Results

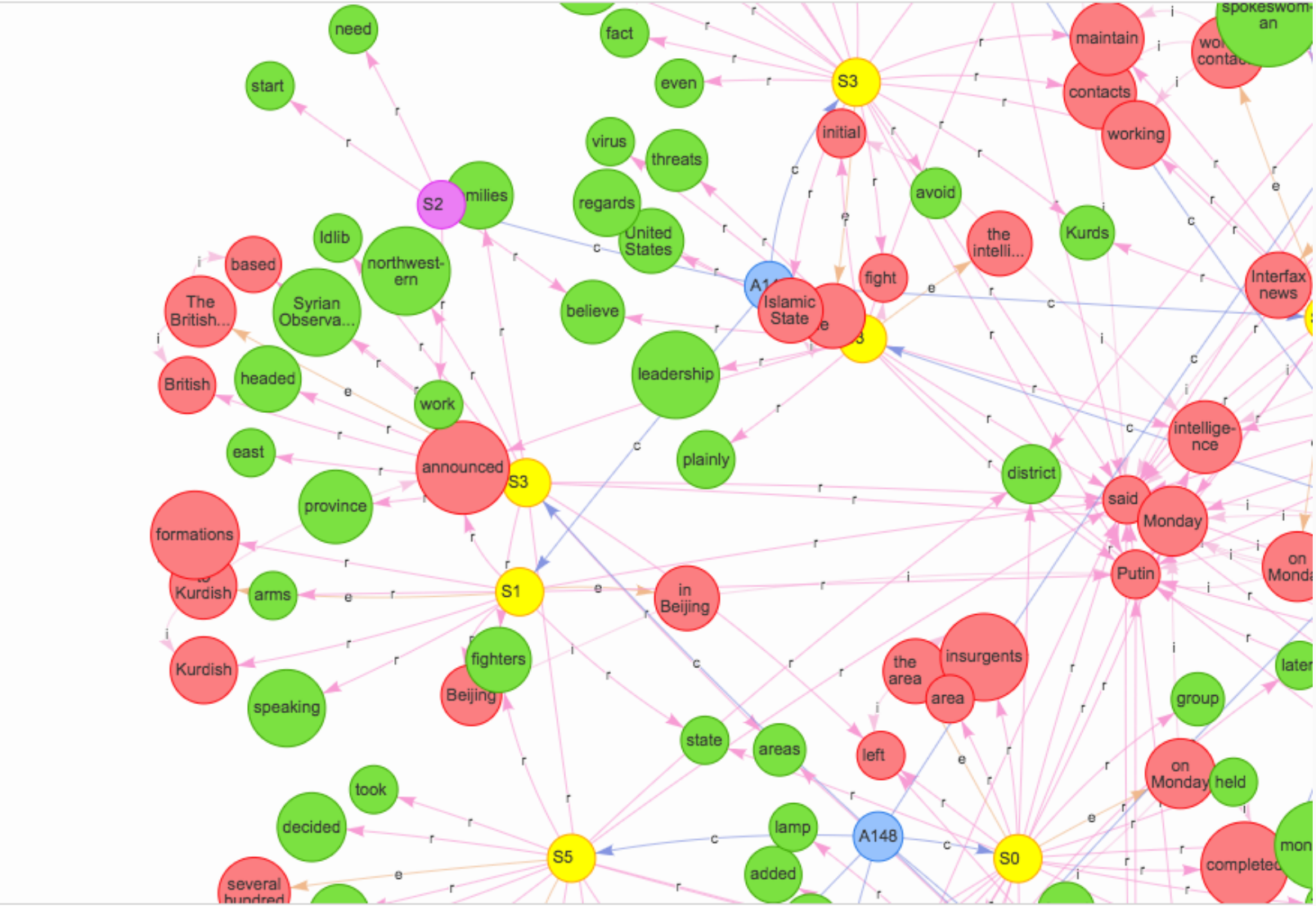
- Content problem used textual information from 124,300 articles from 242 domains
- Structural problem used link information from 19,786 domains (nodes) and 32,632 links (edges)

Model	Bias	Credibility
Content	0.926	0.358
Structure	0.931	0.889

Table 7: Test Set AUC for Bias and Credibility problems. While content is sufficient to detect bias, structure is required to detect fake news.

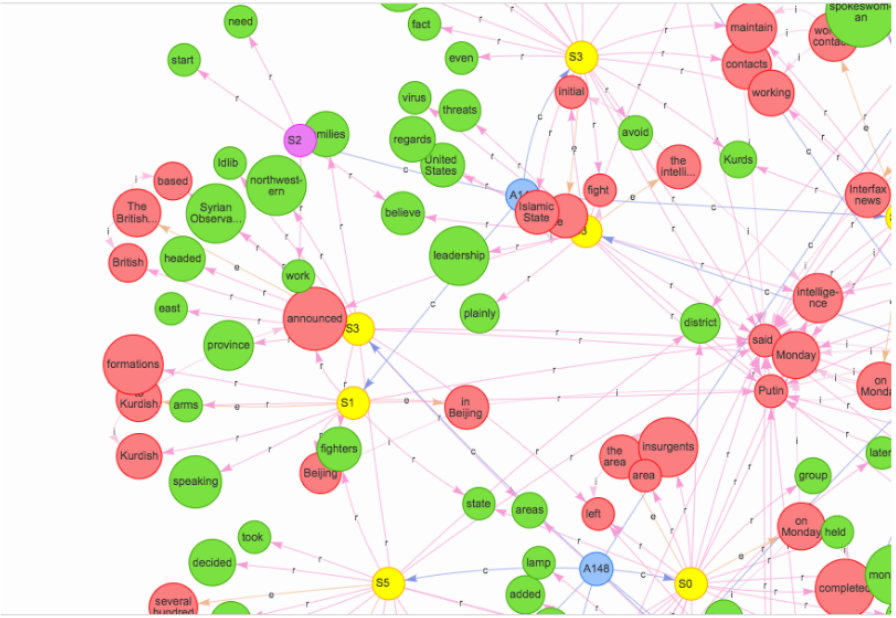
New Directions

The future is knowledge graphs



The future is knowledge graphs

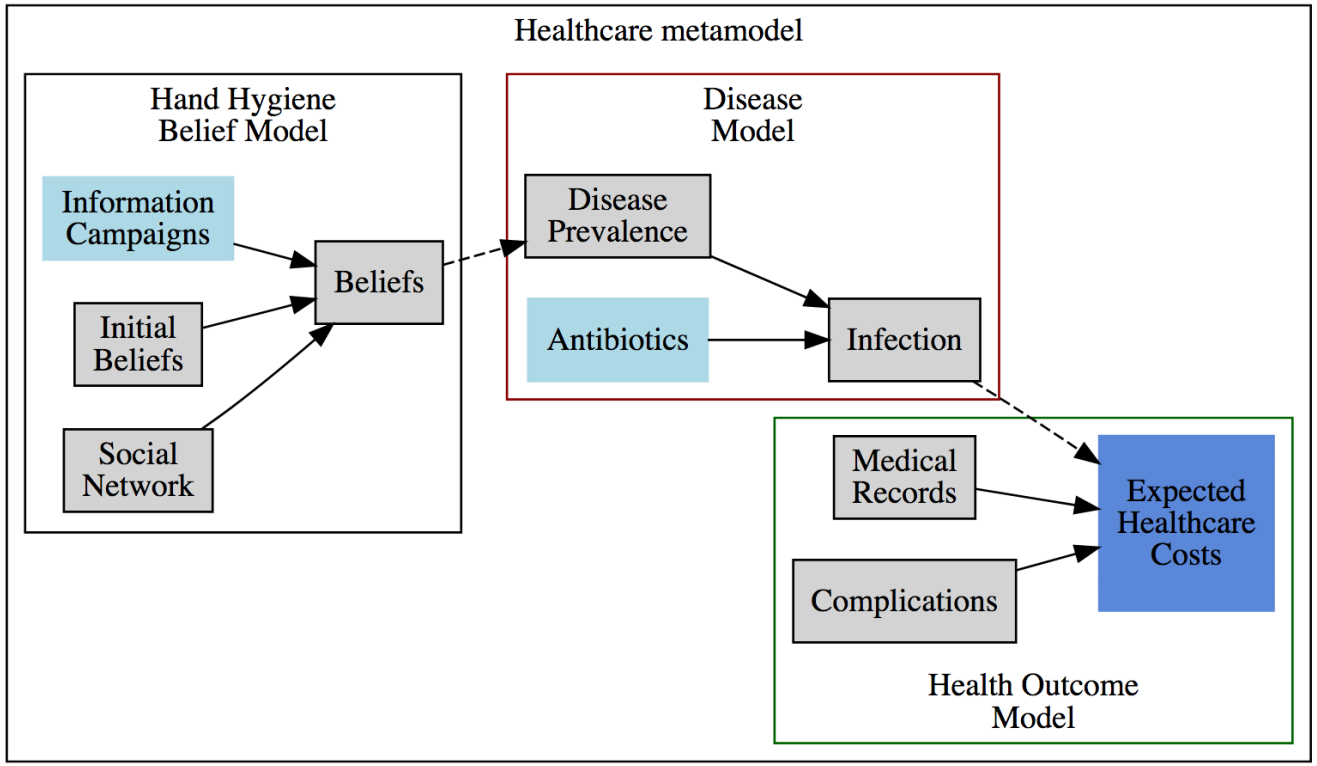
```
{articles(func: has(title)) {
    name
    title
    author {name}
    publication_date
    contains {
      name
      refers {name}}}}
{fact(func: has(establishes)) {
   name
   establishes{name}}}
{inst(func: has(inst)) {
   name
   inst{name}}}
```



Query Languages like SPARQL and GraphQL

Combining Models from Diverse Fields

Understanding the structure of scientific codes in terms of type systems allows for metamodeling



A metamodel for predicting healthcare costs

Figure 20: A hypothetical metamodel for healthcare costs

Reasoning about types – for science!

Reasoning about types help scientists write better code, leveraging this to automate scientific reasoning is the next step

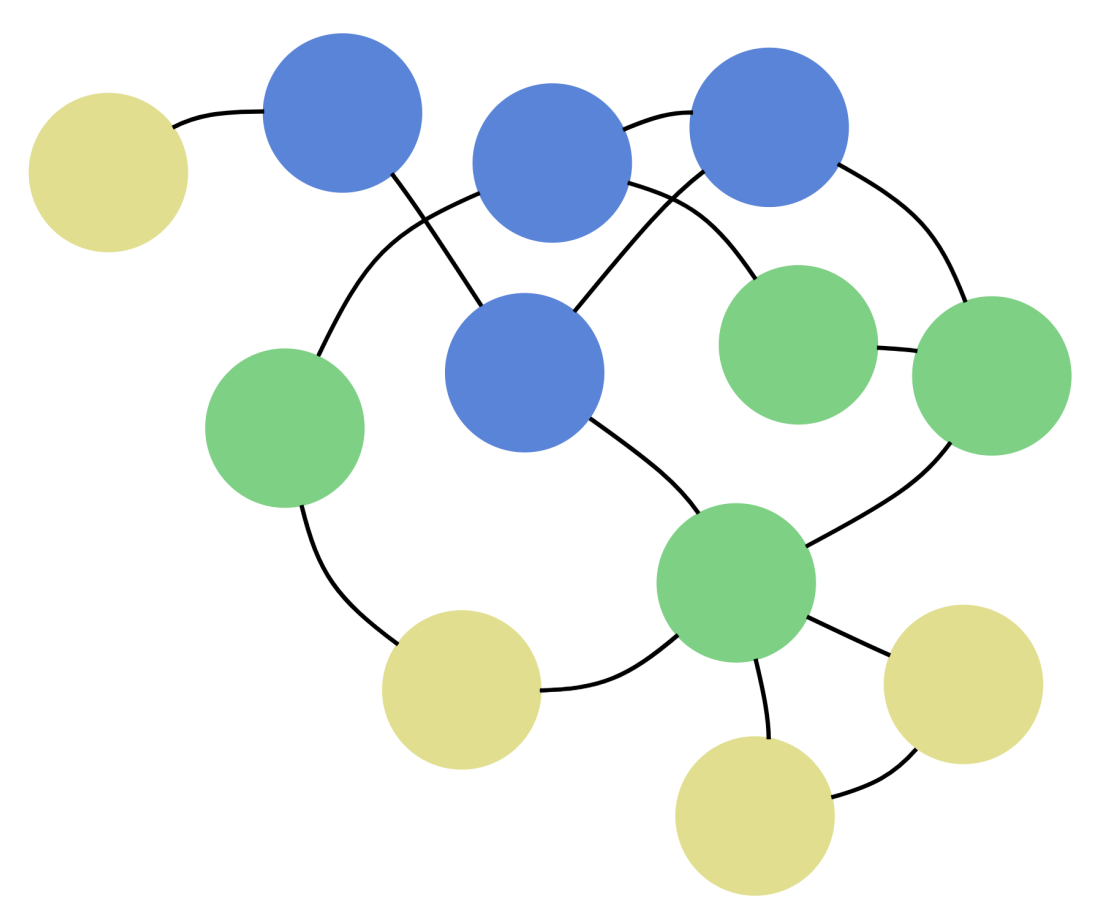


Figure 21: SIR Model graph drawing

Reasoning about types – for science!

Reasoning about types help scientists write better code, leveraging this to automate scientific reasoning is the next step

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dS}{dt} = \frac{\beta SI}{N} - \gamma N$$
$$\frac{dS}{dt} = \gamma N$$

Figure 21: SIR Model expressed as Julia types

```
Beta = Parameter(:β, Unit(Person/Second), TransitionRate())
Gamma = Parameter(:γ, Unit(Person/Second), TransitionRate())
S = Variable(:S, Unit(Person), Amount())
I = Variable(:I, Unit(Person), Amount())
R = Variable(:R, Unit(Person), Amount())
sir = SIR([ODE(Expression(dS/dt -> -βSI/N),
                [Beta], [S]),
                ODE(Expression(dI/dt -> βSI/N -γI),
                    [Beta, Gamma], [S, I]),
            ODE(Expression(dR/dt -> γI), [Gamma], [I])],
           Variable([:N,:S,:I,:R], Unit(Person), Amount()),
           Variable(:t, Unit(Second), Time()))
```

Figure 22: SIR Model as Julia Types

Reasoning about types

- Types are a framework for reasoning about structure
- Put code structure into types – Compiler can go faster
- Put math structure into types – Specialize to exploit structure
- Put model structure into types – Metaprogram on semantic level

Reasoning over Knowledge Graphs

Facts about models and variables become a knowledge graph about scientific modeling

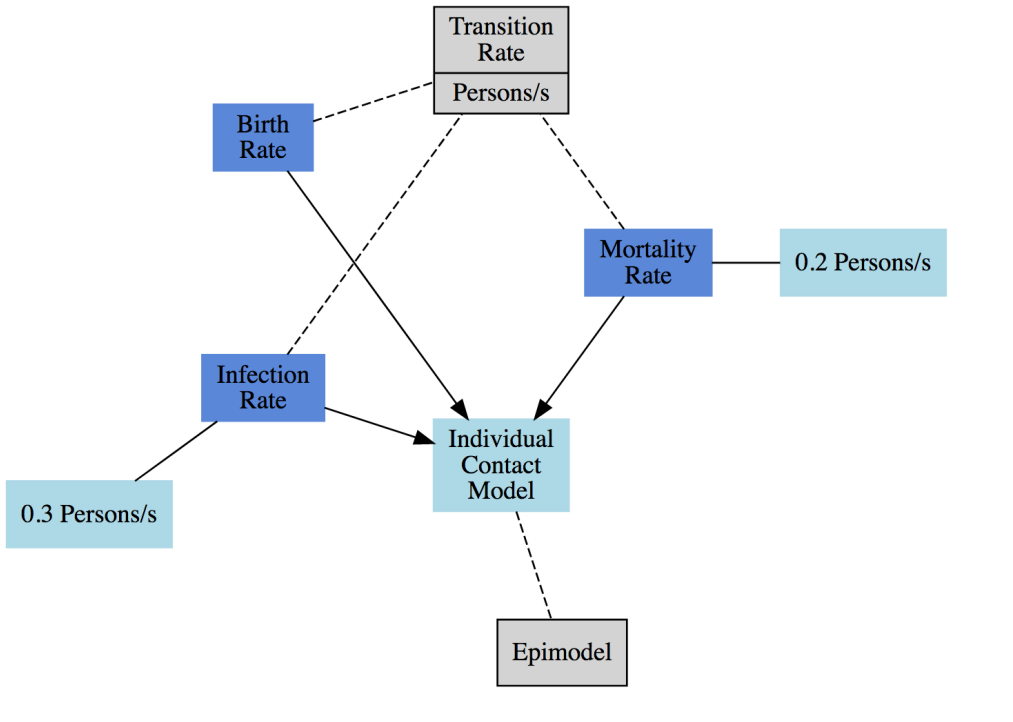


Figure 23: A knowledge graph from epidemiological modeling

Combining Models from Diverse Fields

Understanding the structure of scientific codes in terms of type systems allows for metamodeling

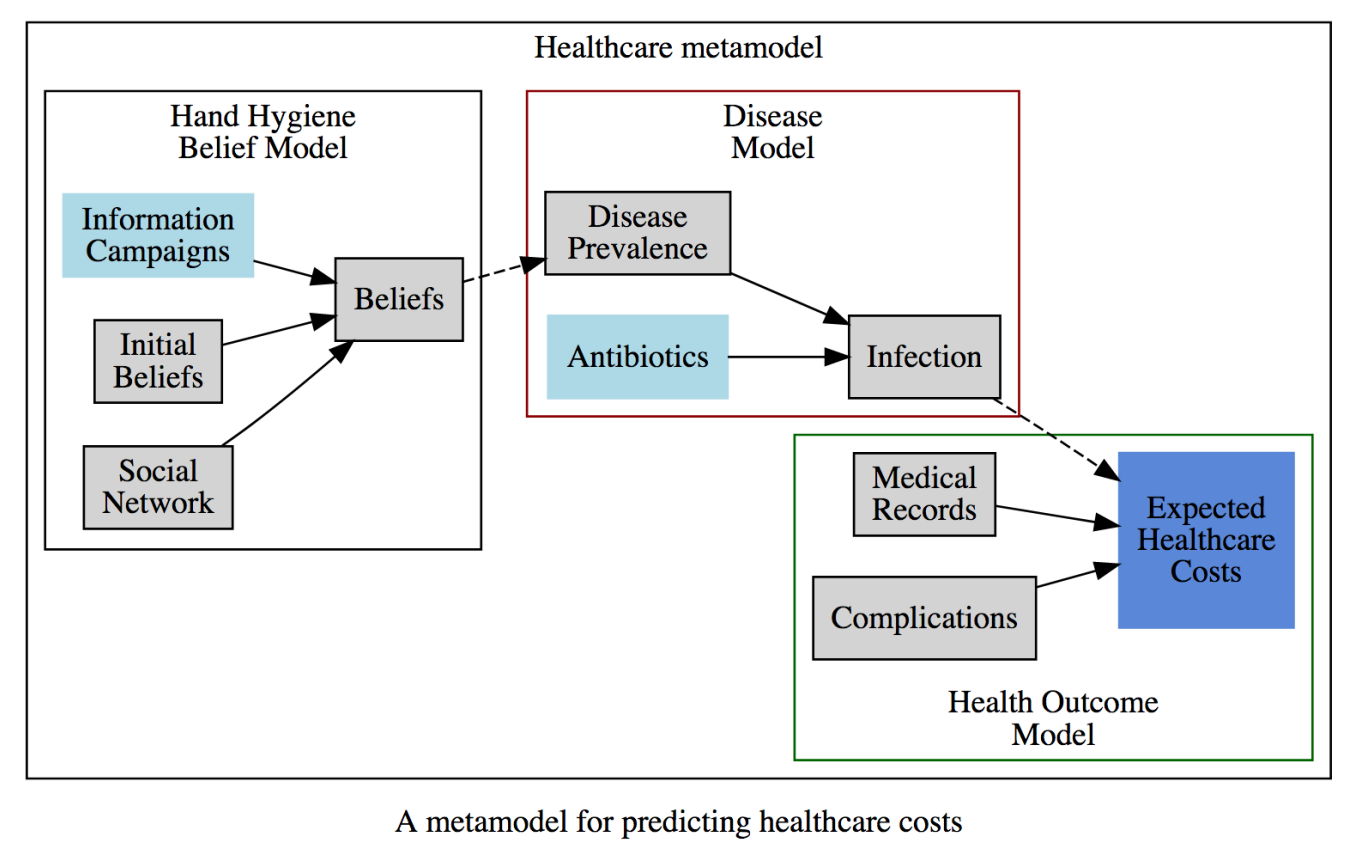


Figure 24: A hypothetical metamodel for healthcare costs

Collaborators

Rohit Varkey Thankachan

Geoff Sanders

Erica Briscoe

Edward Kao

David Bader

Eric Hein

David Ediger

Acknowledgments

DARPA

Office of Naval Research
ONR
Science & Technology
USN

Georgia Tech Research Institute

OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE
UNITED STATES OF AMERICA