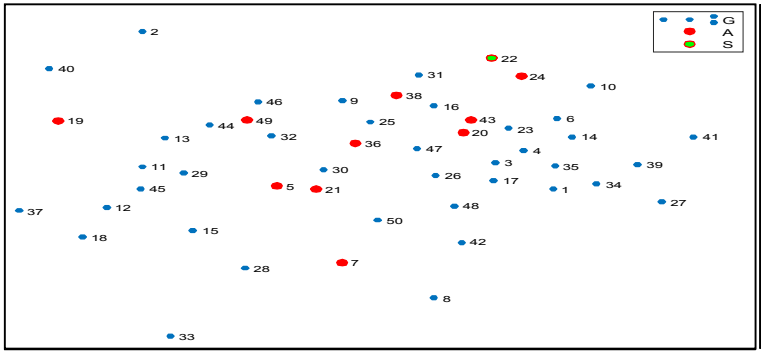
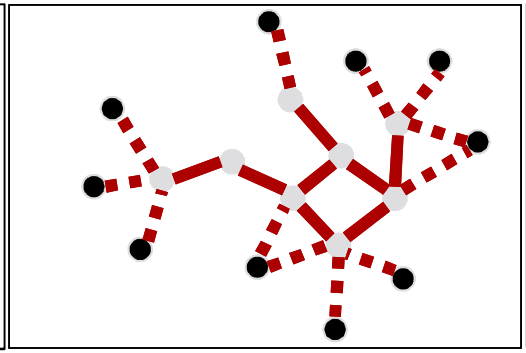
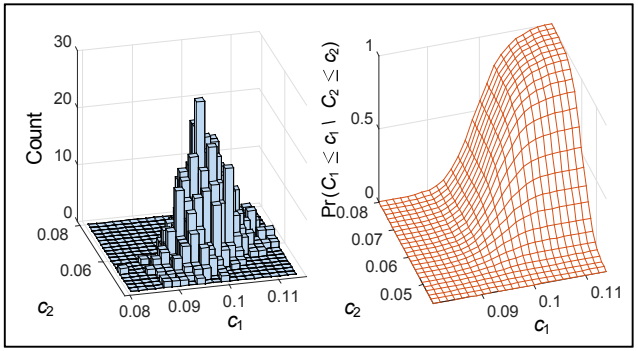
SAND2017-7050PE







Modeling Information Spread Through Large Networks

Rich Field, Jerry Cruz, Tu-Thach Quach, and Jeremy Wendt

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

Sandia National Laboratories

- Motivation
  - Predict information spread through large networks
- Monte Carlo methods
  - Independent cascade (IC)
- Deterministic methods
  - Belief propagation (BP)
- Extending BP to directed graphs
  - Directed belief propagation (DP)
- Applications

Objective and Motivation

- Given a graph
  - Predict information spread
  - Identify the most influential nodes
- Applications
  - Business
  - Elections
  - National security
- Requirements
  - Estimate model parameters from real-world data
  - Results should match real-world data when available
  - Computationally efficient for large networks

Monte Carlo Methods

Sandia National Laboratories

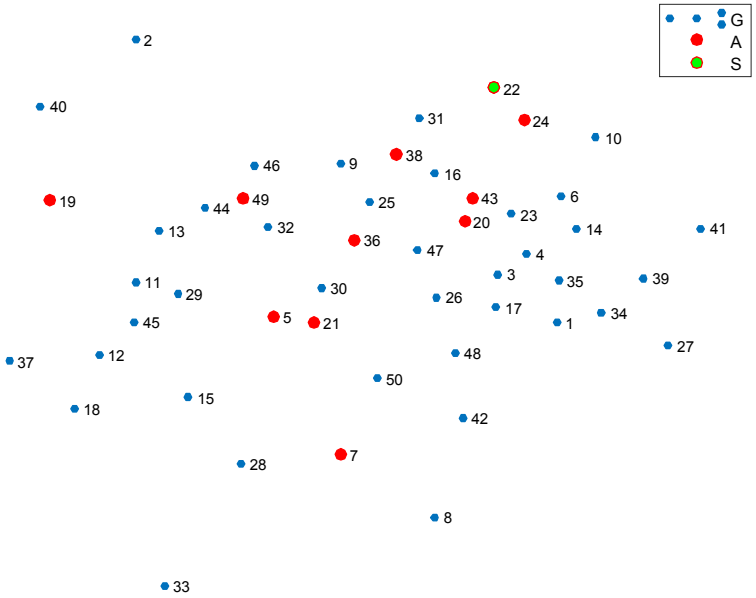
- Pros
  - Simple to understand and implement
  - Applicable to general graphs
  - Time-dependency is represented
  - A variety of metrics can be studied (e.g., joint distributions)
  - Accuracy can be estimated
- Cons
  - Convergence is slow
  - Approach can be infeasible for large graphs
- One approach: Independent cascade (IC)
  - Kempe et al. (2003)
  - Model parameters: influence probabilities

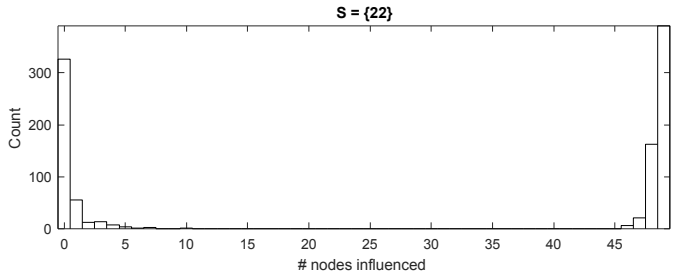
Independent Cascade (IC)

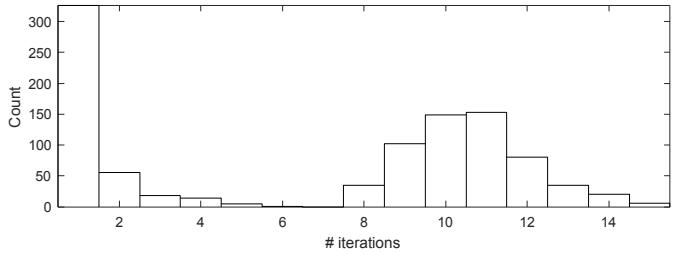
- Let $G(\mathcal{V}, \mathcal{E})$ be a graph with node set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} = \{(i, j),\ 1 \leq i, j \leq n\}$.
- Each node i in the graph has a state $X_i(t) \in \{0, 1\}$ at time t, where $X_i(t) = 1$ and $X_i(t) = 0$ denotes that node i is active and inactive. Node j in the network exhibits influence on its neighbors, that is, on the set of nodes $\mathcal{N}(j) = \{i \in \mathcal{V}, i \neq j : (i, j) \in \mathcal{E}\}$.
- We say that node $i \in \mathcal{V}$ is "contagious" at time t if $X_i(t) = 1$ and $X_i(t - 1) = 0$, that is, if it is active at time t but was inactive at time $t - 1$.
- Assuming node i is contagious at time t, it attempts to influence its neighbors $j \in \mathcal{N}(i)$ with probability $p(i, j) \in [0, 1]$. If successful, node j becomes active at time $t + 1$; otherwise node j is not active. At time $t + 1$, node i is no longer contagious and is unable to influence any more nodes.
- To start the spread of influence through G, there must be an initial condition. Let $S \subset \mathcal{V}$ be a collection of "seed" nodes that are assumed contagious at time $t = 0$. At each time step, the contagious nodes are allowed to activate their neighboring nodes with probabilities $p(i, j) \in [0, 1]$. The simulation ends when no additional node become contagious or all nodes have been activated.

IC Example

- E-R graph with n = 50 nodes and $p(i,j)$ = 0.1. Seed node S = 22 (green).
- At convergence, 11 (red) nodes influenced; 39 (blue) nodes remain inactive
- IC model is repeated 1,000 times with identical seed node
- Distribution of number of nodes influenced (top)
- Number of iterations of IC needed for convergence







A Deterministic Method

Sandia National Laboratories

- Belief propagation (BP)
  - J. Pearl (1982), (Yedidia, 2001)
  - Adaptable to iterative solver for linear systems
- Pros
  - Solve for marginal distributions directly
  - Minimal computational effort
- Cons
  - More difficult to understand and implement
  - Strict assumptions on graph structure (e.g., polytree)
  - Used in general graphs (“loopy BP”); consequences of this are largely unknown
  - Not really appropriate for directed graphs
- Model parameters
  - Initial beliefs and potential functions

Belief Propagation (BP)

- Let $G(\mathcal{V}, \mathcal{E})$ be a graph with node set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} = \{(i, j),\ 1 \leq i, j \leq n\}$. We will assume G contains no directed cycles, that is, there exist no closed paths in the graph.

- Let $\phi_i(x_i)$ quantify node i's belief it has value x_i; this quantity represents previous knowledge about node i by, for example, evidence gained from observations. Further, let $\psi_{ij}(x_i, x_j)$ quantify the probability that node j believes it has value x_j given that node i believes it has value x_i. Then

$$m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i)\, \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus \{j\}} m_{ki}(x_i)$$

  denotes a message passed from node i to node j describing node i's belief that node j has value x_j, where $\mathcal{N}(j) = \{i \in \mathcal{V}, i \neq j \colon (i, j) \in \mathcal{E}\}$ defines the neighborhood of node j.

- The marginal probability at node k is then the product of all incoming messages at that node multiplied by its prior belief, i.e.,

$$p(x_k) \propto \phi_k(x_k) \prod_{j \in \mathcal{N}(k)} m_{jk}(x_k)$$

Some Additional Details

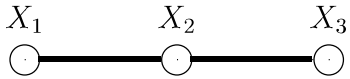
- Assume the state of each node is $x_i \in \{0, 1\}$.
- $\phi_i(x_i)$ is a 2×1 vector representing the initial belief that node i is in state 0 or 1.
  - If the state of node i is known, this is generally $(0.999, 0.001)^T$.
  - If the state of node i is unknown, this is generally set to $(0.5, 0.5)^T$.
- $\psi_{ij}(x_i, x_j)$ is a 2×2 matrix representing the probability that node j believes it has value x_j given that node i believes it has value x_i, for all possible values of x_i and x_j.

$$\text{Homophily: } \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix} \qquad \text{Heterophily: } \begin{bmatrix} 0.01 & 0.99 \\ 0.99 & 0.01 \end{bmatrix}$$

- $\prod_{k \in \mathcal{N}(i) \setminus \{j\}}$ ensures that node j's message to node i don't come right back.
- Message passing repeats until convergence is attained.

BP Example

- Consider a simple graph with 3 nodes, 2 edges



- We assume random variables x_1, x_2, and x_3 each take values in {0, 1} with potential functions

$$\psi_{1,2}(x_1, x_2) = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} \quad \text{and} \quad \psi_{2,3}(x_2, x_3) = \begin{pmatrix} 0.1 & 1 \\ 1 & 0.1 \end{pmatrix}$$

and initial beliefs

$$\phi_1(x_1) = \phi_3(x_3) = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \quad \text{and} \quad \phi_2(x_2) = \begin{pmatrix} 1 \\ 0.1 \end{pmatrix}$$

  - Node 1 has a mild preference to be like node 2
  - Node 3 strongly wants to be the opposite of node 2
  - Node 2 has strong initial belief that it is in state 0
  - Nodes 1 and 3 initially believe they are in state 0 or 1 with equal probability

Messages

$$m_{12}(x_2) = \sum_{x_1} \phi_1(x_1)\,\psi_{1,2}(x_1, x_2) \prod_{k \in \mathcal{N}(1)\setminus\{2\}} m_{k1}(x_1) = \sum_{x_1} \phi_1(x_1)\,\psi_{1,2}(x_1, x_2)$$

$$= 0.5 \begin{pmatrix} 1 \\ 0.9 \end{pmatrix} + 0.5 \begin{pmatrix} 0.9 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.95 \\ 0.95 \end{pmatrix}$$

$$m_{32}(x_2) = \sum_{x_3} \phi_3(x_3)\,\psi_{3,2}(x_3, x_2) \prod_{k \in \mathcal{N}(3)\setminus\{2\}} m_{k3}(x_3) = \sum_{x_3} \phi_3(x_3)\,\psi_{3,2}(x_3, x_2)$$

$$= 0.5 \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} + 0.5 \begin{pmatrix} 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.75 \\ 0.75 \end{pmatrix}$$

$$m_{21}(x_1) = \sum_{x_2} \phi_2(x_2)\,\psi_{2,1}(x_2, x_1) \prod_{k \in \mathcal{N}(2)\setminus\{1\}} m_{k2}(x_2) = \sum_{x_2} \phi_2(x_2)\,\psi_{2,1}(x_2, x_1)\,m_{32}(x_2)$$

$$= 1.0 \begin{pmatrix} 1 \\ 0.9 \end{pmatrix} (0.75) + 0.1 \begin{pmatrix} 0.9 \\ 1 \end{pmatrix} (0.75) = \begin{pmatrix} 0.8175 \\ 0.75 \end{pmatrix}$$

$$m_{23}(x_3) = \sum_{x_2} \phi_2(x_2)\,\psi_{2,3}(x_2, x_3) \prod_{k \in \mathcal{N}(2)\setminus\{3\}} m_{k2}(x_2) = \sum_{x_2} \phi_2(x_2)\,\psi_{2,3}(x_2, x_3)\,m_{12}(x_2)$$

$$= 1.0 \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} (0.95) + 0.1 \begin{pmatrix} 1 \\ 0.1 \end{pmatrix} (0.95) = \begin{pmatrix} 0.19 \\ 0.9595 \end{pmatrix}$$

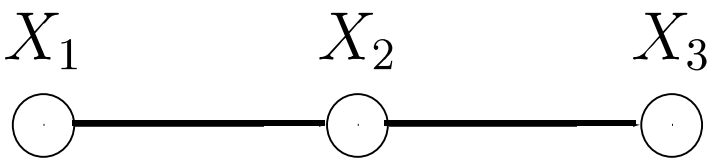
Final Result

- Marginal probabilities for the state of each node

$$p(x_1) \propto \phi_1(x_1)\, m_{21}(x_1) = \begin{pmatrix} 0.40875 \\ 0.375 \end{pmatrix} \quad \Longrightarrow \quad p(x_1) \approx \begin{pmatrix} 0.522 \\ 0.478 \end{pmatrix}$$

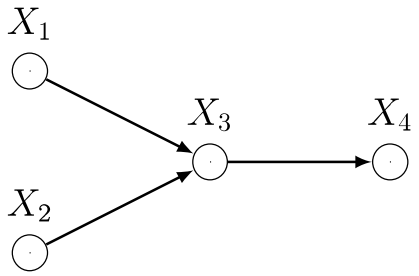
$$p(x_2) \propto \phi_2(x_2)\, m_{12}(x_2)\, m_{32}(x_2) = \begin{pmatrix} 0.7125 \\ 0.07125 \end{pmatrix} \quad \Longrightarrow \quad p(x_2) \approx \begin{pmatrix} 0.909 \\ 0.091 \end{pmatrix}$$

$$p(x_3) \propto \phi_3(x_3)\, m_{23}(x_3) = \begin{pmatrix} 0.095 \\ 0.47975 \end{pmatrix} \quad \Longrightarrow \quad p(x_3) \approx \begin{pmatrix} 0.165 \\ 0.835 \end{pmatrix}$$



BP Extended to Digraphs

- A modification of Belief Propagation that preserves directed influence
  - BP passes update messages in both directions along edges in a graph
  - In directed propagation, messages pass only downstream
- Requires per-node and per-edge functions
  - Per-node may be learned from real-world data
  - Per-edge based on node in-degree
- Results in each node's likelihood of adoption
  - Diffusion score is sum of all nodes' likelihoods
- Implementation details in the paper
  - Available at https://github.com/algorithmfoundry/Foundry
  - Details on BP (Yedidia, 2001)



Directed BP – Standard BP on a Collection of Subgraphs

Sandia National Laboratories

- Let $G = (\mathcal{V}, \mathcal{E})$ be a directed tree or polytree with node set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} = \{(i, j) \colon 1 \leq i, j \leq n\}$.
- Suppose a directed path of two or more nodes exists from node $u \in \mathcal{V}$ to node $v \in \mathcal{V}$ which we denote by $u \to v$; because G is assumed a polytree we know each $u \to v$ path is unique.
- We define a collection of subgraphs $G_k = (\mathcal{V}_k, \mathcal{E}_k) \subset G$, $k = 1, \ldots, n$, where

$$\mathcal{V}_k = \{k\} \cup \{u \in \mathcal{V} \colon \text{there exists a path } u \to k\}$$
$$E_k = \{(i, j) \colon i, j \in \mathcal{V}_k \text{ and } (i, j) \in \mathcal{E}\}$$

  denote the node set and edge set, respectively, of subgraph G_k.

More on DP

- Then

$$f_k(x_k) = \sum_{x_j \,:\, j \in \mathcal{V}_k \setminus \{k\}} p(\mathbf{x}_{\mathcal{V}_k})$$

is the marginal pdf of node k with respect to graph G_k, which can be related to the marginal pdf with respect to the original graph G via

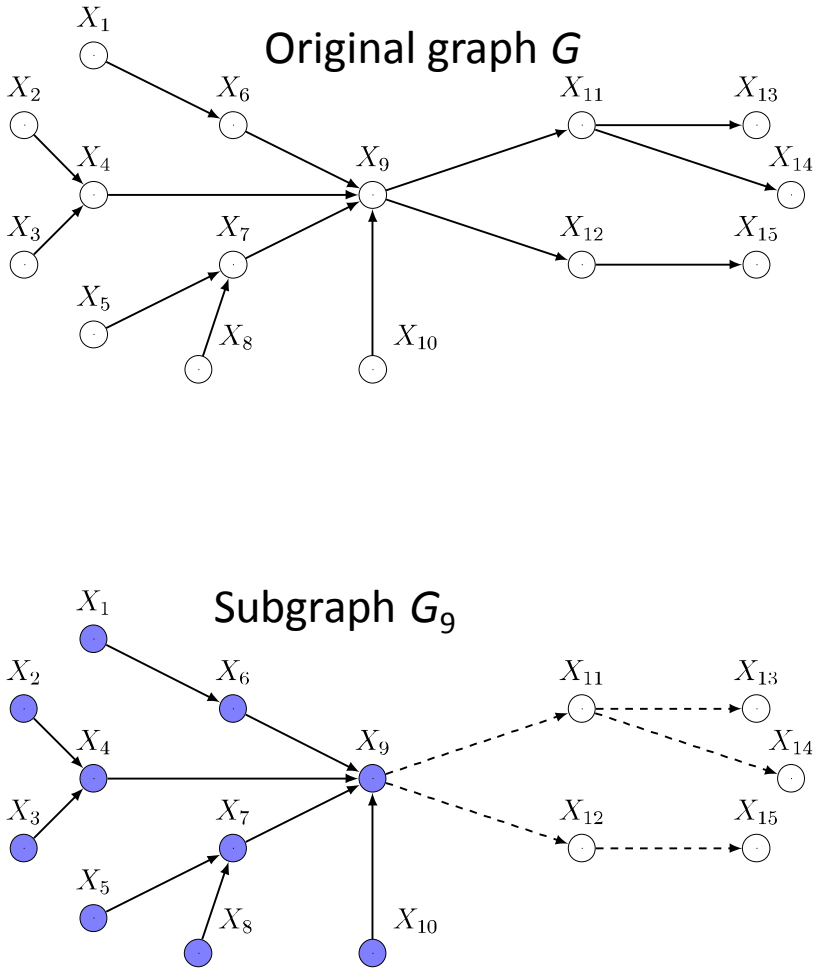
$$p(x_k) = f_k(x_k) \sum_{x_j \,:\, j \in \mathcal{V} \setminus \mathcal{V}_k \setminus \{k\}} p(\mathbf{x}_{\mathcal{V} \setminus \mathcal{V}_k})$$

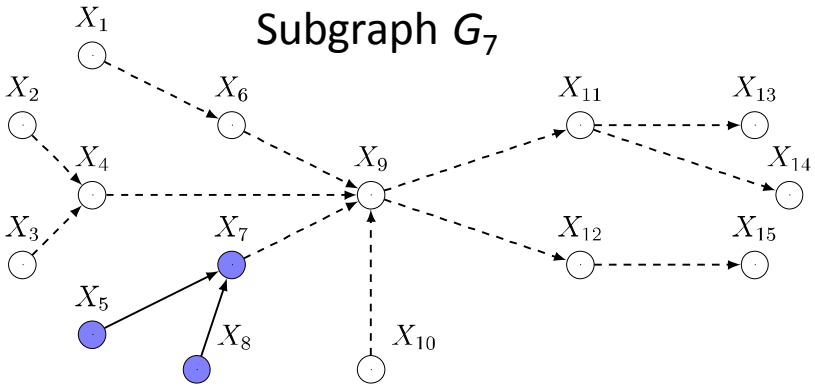
- Then, assuming nodes $i, j \in \mathcal{V}_k$,

$$m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i)\, \psi_{i,j}(x_i, x_j) \prod_{k \in (\mathcal{N}(i) \cap \mathcal{V}_k) \setminus \{j\}} m_{ki}(x_i)$$

denotes a message passed from node i to node j on subgraph G_k describing node i's belief that node j has value x_j. Only those nodes that can have influence on node j are considered; this is equivalent to computing only the "forward messages" to node j, and ignoring any "backwards messages".

An Example


Original graph G
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
X_9
X_{10}
X_{11}
X_{12}
X_{13}
X_{14}
X_{15}
Subgraph G_9
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
X_9
X_{10}
X_{11}
X_{12}
X_{13}
X_{14}
X_{15}


Subgraph G_7
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
X_9
X_{10}
X_{11}
X_{12}
X_{13}
X_{14}
X_{15}

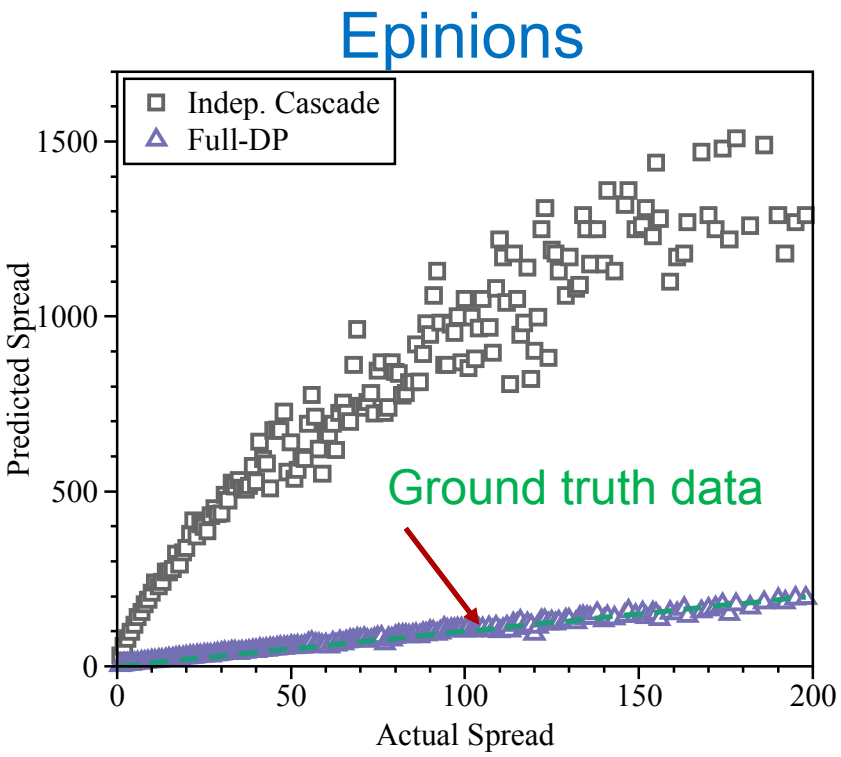
Application #1 – Influence Spread

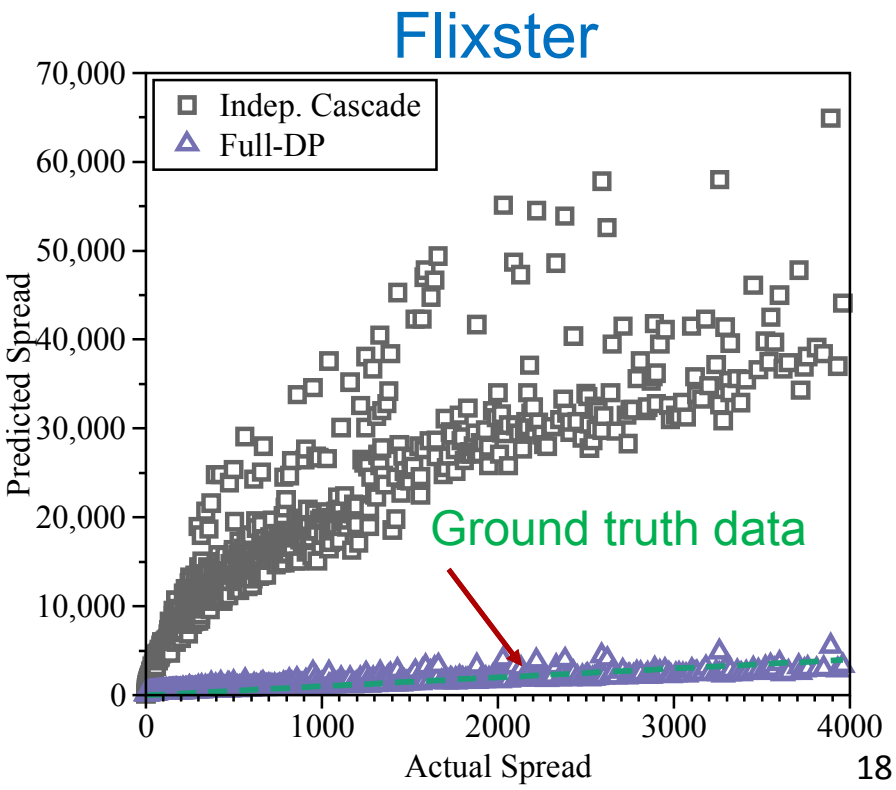
Sandia National Laboratories

- Data sets
  - Flixster movie review propagation
    - 800K nodes; 12M edges
  - Epinions product review propagation
    - 18K nodes; 1.2M edges
- Diffusion algorithms
  - Independent cascade (IC)
  - Directed belief propagation (DP)
    1. Degree-weight per-edge; learned per-node (Full-DP)
    2. Degree-weight per-edge; constant per-node (Edge-DP)
    3. Constant-weight per-edge; learned per-node (Node-DP)

Predicting Influence Spread (1 of 2)

- Directed propagation (DP) matches real-world data fairly well
- Independent cascade (IC) does not

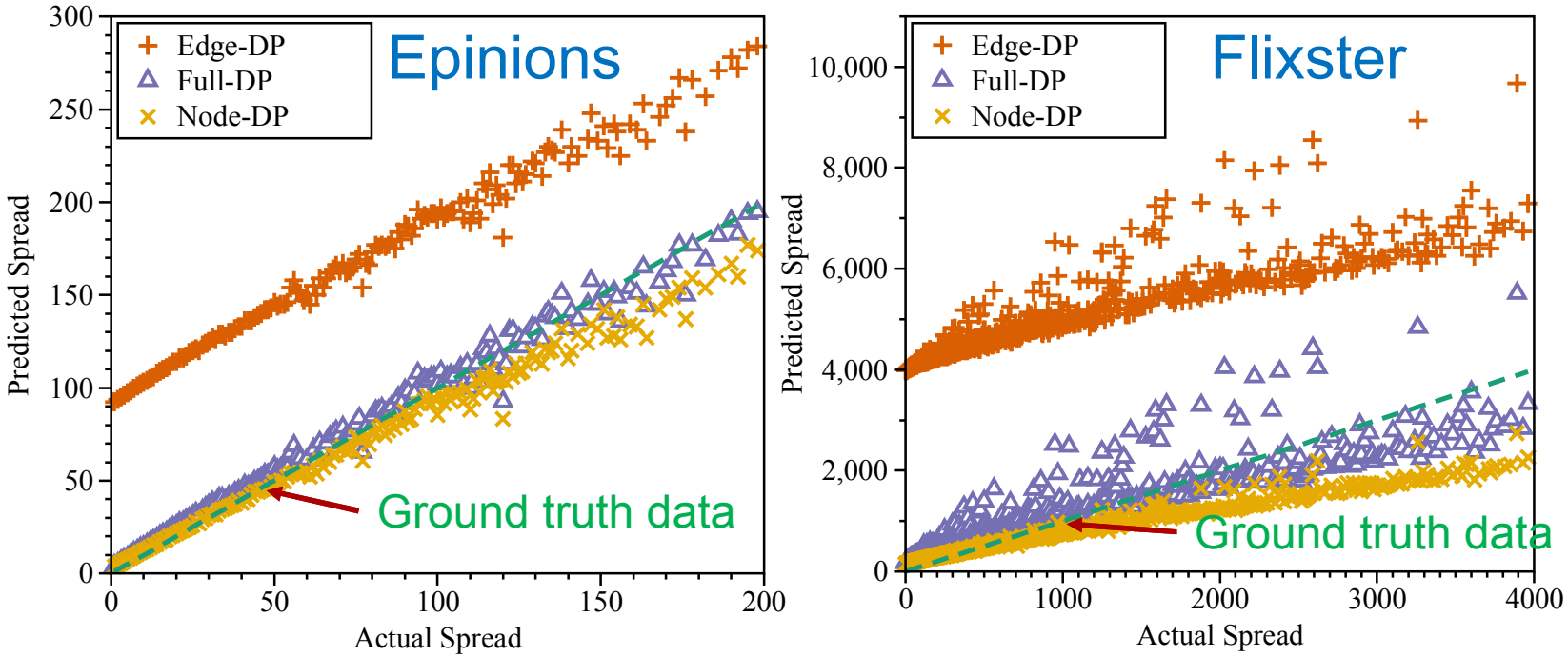




Predicting Influence Spread (2 of 2)

Sandia National Laboratories

- Three versions of DP
  1. Degree-weight per-edge; learned per-node (Full-DP)
  2. Degree-weight per-edge; constant per-node (Edge-DP)
  3. Constant-weight per-edge; learned per-node (Node-DP)
- Performs best when trained with minimal real-world data

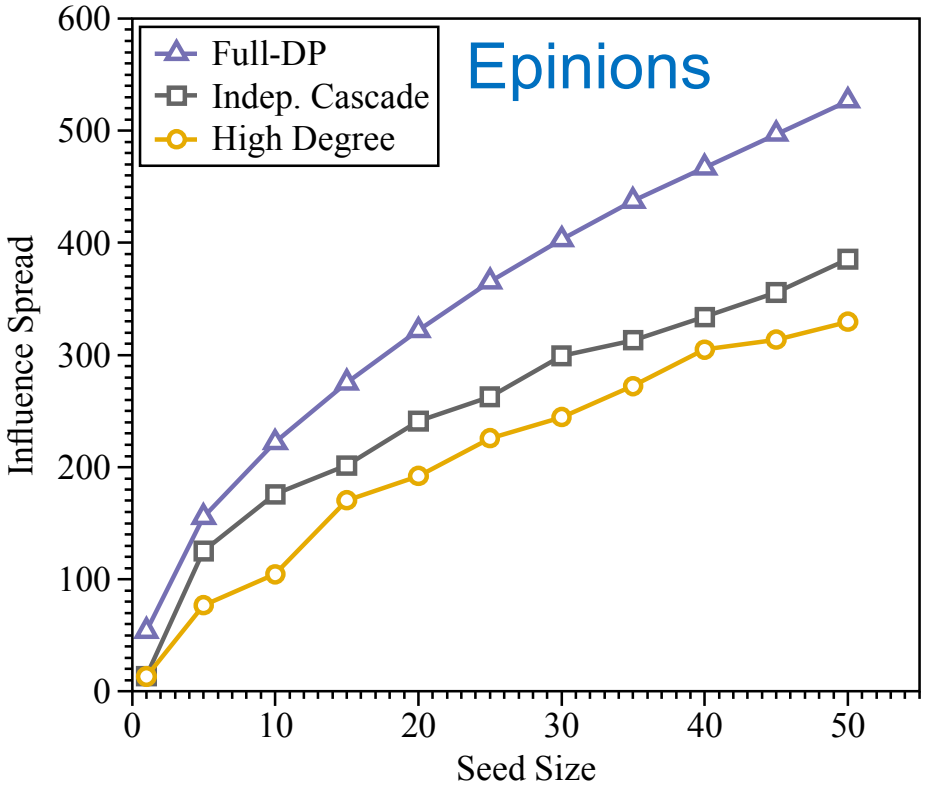


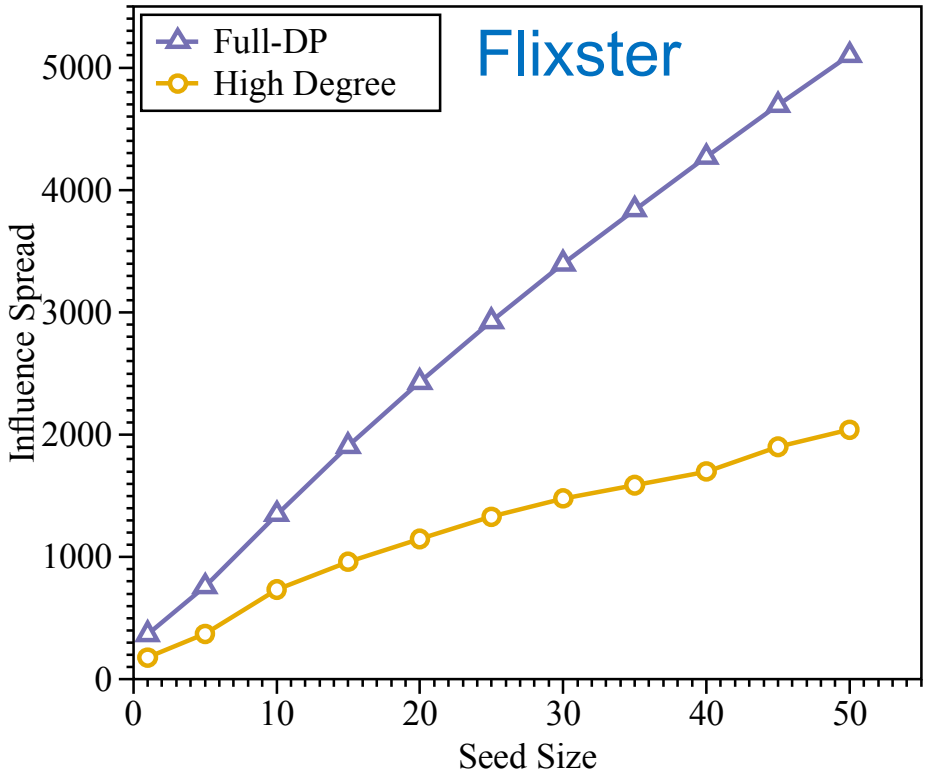
Application #2 – Maximal Influencers

- Full *k*-seed influence maximization is NP-Hard
  - Greedy algorithm widely used (Kempe, 2003)
  - CELF gives same set; more efficient (Leskovec, 2007)
- Diffusion algorithms tested
  - IC (Epinions only)
  - Full-DP
  - Edge-DP
  - High Degree*
- Procedure
  - For given diffusion method, compute 50 most influential nodes using CELF
  - For common comparison, compute diffusion spread for those seeds using Full-DP

*High Degree selects nodes based on degree – does not require CELF runs

Influence Maximization (1 of 2)





Note: Edge-DP not shown as results are nearly identical to Full-DP

Influence Maximization (2 of 2)

- Overlap between identified seed sets

Epinions	10	20	30	40	50
High Degree	3	6	10	16	18
Edge-DP	10	18	29	36	45
Indep. Cascade	6	9	14	18	24

Flixster	10	20	30	40	50
High Degree	0	1	3	4	5
Edge-DP	10	20	30	40	49

- How do the selected seeds differ?
  - *Community Detection:* Full-DP and IC chose seeds in separate communities more than Max Degree
  - *Average Distance:* Full-DP chose seeds further apart than IC which chose further apart than Max Degree
  - *Node Degree:* Full-DP chose lower degree nodes than IC which chose lower degree nodes than Max Degree
    - Full-DP and IC chose nodes well above average degree
- Balance between higher degree and distance between seeds

Compute Resources

Sandia National Laboratories

- Full DP (Flixster maximization)
  - Initial computation for each node as seed: 12 hours on 60 compute nodes
  - CELF identification of 50 top nodes: 16 minutes on workstation
  - Average propagation: 4 seconds
    - Contrast IC with 10,000 MC simulations: 6 minutes

Summary and Conclusions

- Objective: Predict information spread through large networks
- Monte Carlo methods
  - Independent cascade
- Deterministic methods
  - Belief propagation (BP)
  - Extension: Directed belief propagation (DP)
- Applications
  - Influence spread
  - Influence maximization
- BP and DP
  - More accurate to real-world datasets
  - Computationally efficient enough for large-scale datasets