Cost-effective Outbreak Detection in Networks

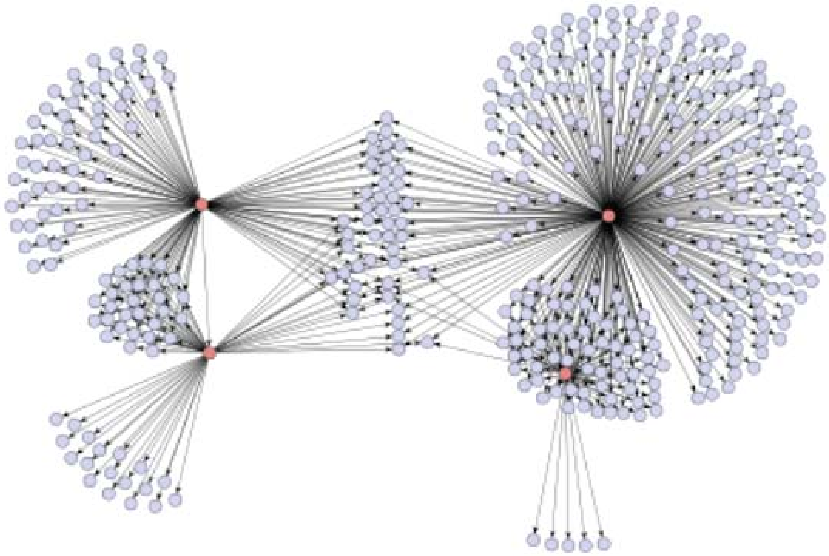
Jure Leskovec

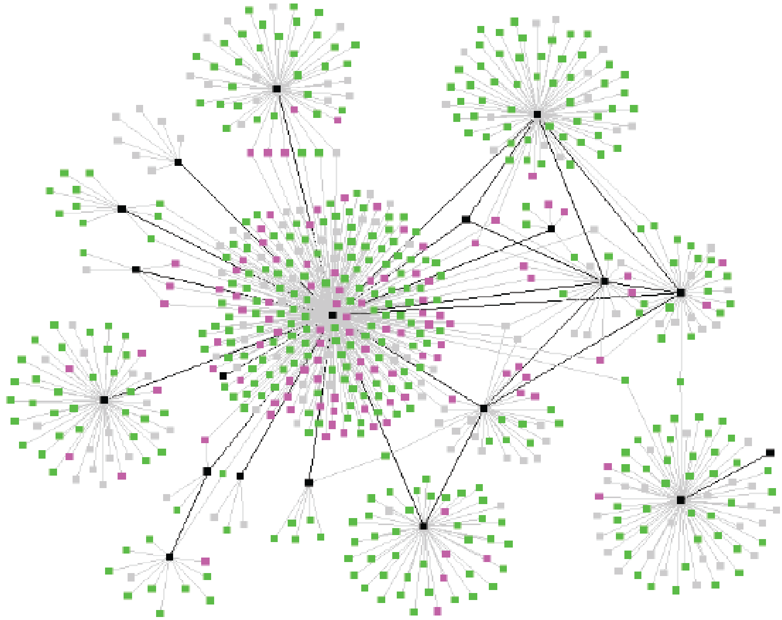
Joint work with Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance

Carnegie Mellon

Nielsen BuzzMetrics

Diffusion in Social Networks





- One of the networks is a spread of a disease, the other one is product recommendations
- Which is which? ☺

Diffusion in Social Networks

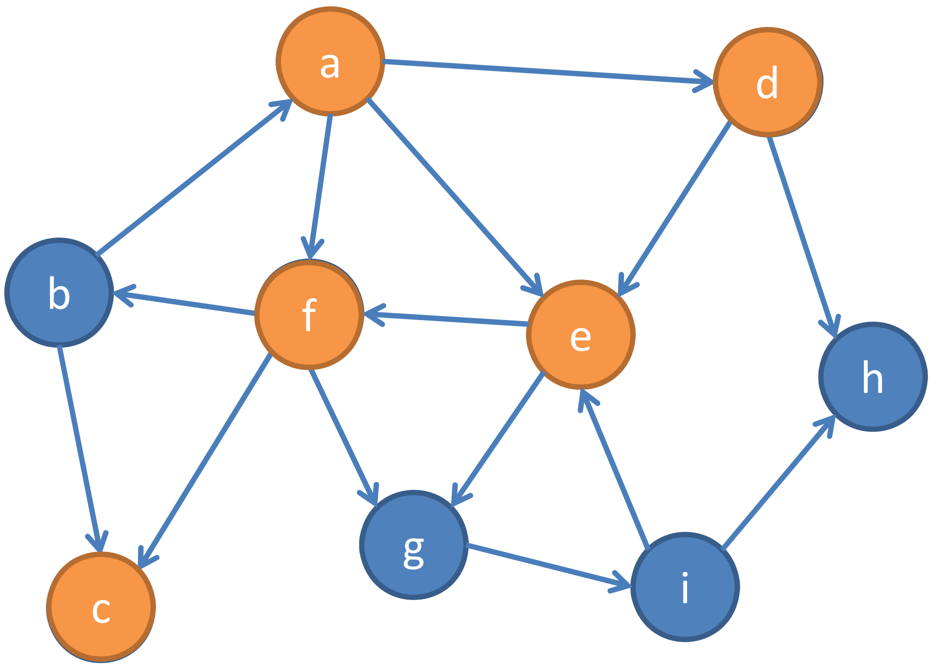
- A fundamental process in social networks:
  Behaviors that cascade from node to node like an epidemic
  - News, opinions, rumors, fads, urban legends, …
  - Word-of-mouth effects in marketing: rise of new websites, free web based services
  - Virus, disease propagation
  - Change in social priorities: smoking, recycling
  - Saturation news coverage: topic diffusion among bloggers
  - Internet-energized political campaigns
  - Cascading failures in financial markets
  - Localized effects: riots, people walking out of a lecture

Empirical Studies of Diffusion

- Experimental studies of diffusion have long history:
  - Spread of new agricultural practices [Ryan-Gross 1943]
    - Adoption of a new hybrid-corn between the 259 farmers in Iowa
    - Classical study of diffusion
    - Interpersonal network plays important role in adoption
      → Diffusion is a social process
  - Spread of new medical practices [Coleman et al 1966]
    - Studied the adoption of a new drug between doctors in Illinois
    - Clinical studies and scientific evaluations were not sufficient to convince the doctors
    - It was the social power of peers that led to adoption

Diffusion in Networks

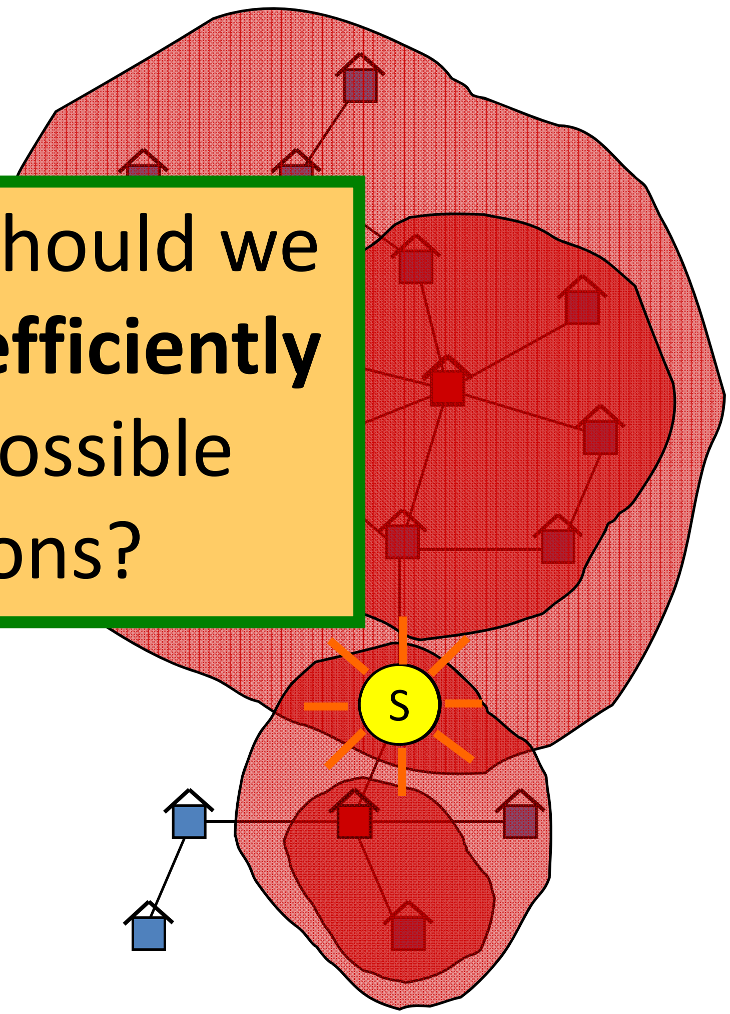
- Initially some nodes are active
- Active nodes spread their influence on the other nodes, and so on …



Scenario 1: Water Network

- Given a real city water distribu
- And dat contam the net
- Problen *Environmental Protection Agency*

On which nodes should we place **sensors** to **efficiently** detect the all possible contaminations?



Scenario 2: Online media

Which news websites should one read to **detect new stories** as **quickly** as possible?

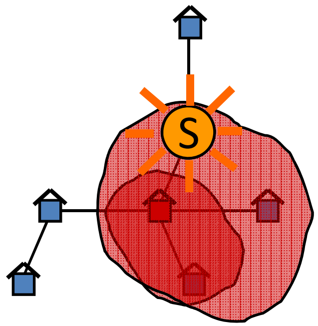
Cascade Detection: General Problem

- Given a dynamic process spreading over the network
- We want to select a set of nodes to detect the process effectively
- Many other applications:
  - Epidemics
  - Network security

Two Parts to the Problem

- Reward, *e.g.*:
  - 1) Minimize time to detection
  - 2) Maximize number of detected propagations
  - 3) Minimize number of infected people
- Cost (location dependent):
  - Reading big blogs is more time consuming
  - Placing a sensor in a remote location is expensive

Problem Setting



- Given a graph $G(V,E)$
- and a budget B for sensors
- and data on how contaminations spread over the network:
  - for each contamination i we know the time $T(i, u)$ when it contaminated node u
- Select a subset of nodes A that maximize the expected reward

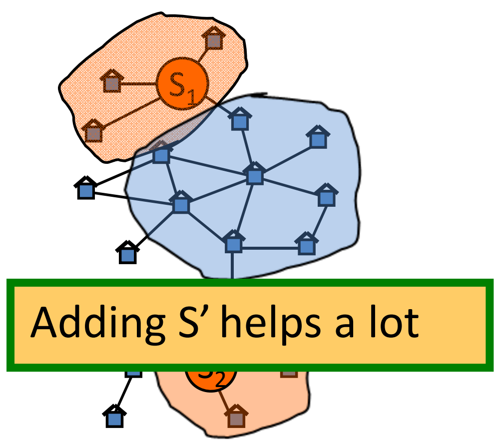
$$\max_{\mathcal{A}\subseteq\mathcal{V}} R(\mathcal{A}) \equiv \sum_i P(i) \underbrace{R_i(T(i,\mathcal{A}))}_{\text{Reward for detecting contamination } i}$$

subject to $cost(A) < B$

Structure of the Problem

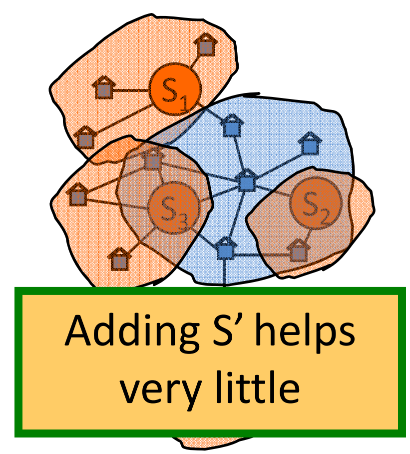
- Solving the problem exactly is NP-hard
  - Set cover (or vertex cover)

- Observation: Diminishing returns



Placement A={S_1, S_2}

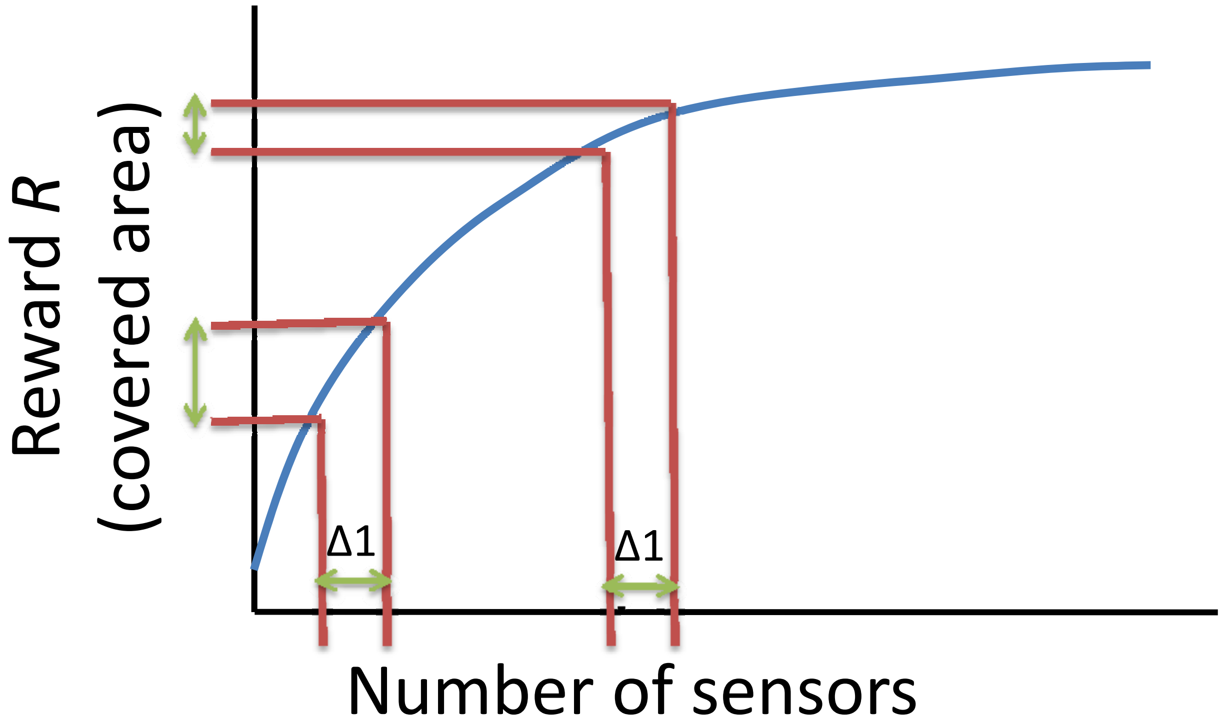
New sensor:



Placement B={S_1, S_2, S_3, S_4}

Analysis

- <u>Analysis: </u>diminishing returns at individual nodes implies diminishing returns at a "global" level
  - Covered area grows slower and slower with placement size



An Approximation Result

- Diminishing returns: Covered area grows slower and slower with placement size

R is submodular: if $A \subseteq B$ then

$$R(A \cup \{x\}) - R(A) \geq R(B \cup \{x\}) - R(B)$$

Theorem [Nehmhauser et al. '78]:

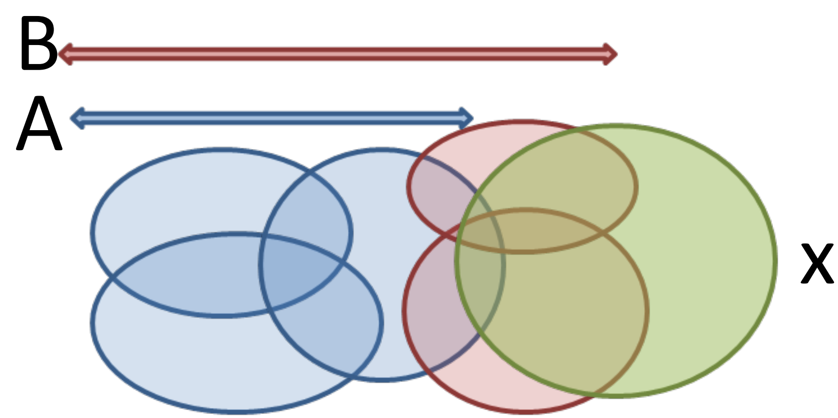
If *f* is a function that is monotone and submodular, then *k*-step hill-climbing produces set *S* for which *f(S)* is within *(1-1/e)* of optimal.

Reward functions: Submodularity

- We must show that R is submodular:

$$\underbrace{R(\mathcal{A} \cup \{s\}) - R(\mathcal{A})}_{\text{Benefit of adding a sensor to a small placement}} \geq \underbrace{R(\mathcal{B} \cup \{s\}) - R(\mathcal{B})}_{\text{Benefit of adding a sensor to a large placement}}$$

- What do we know about submodular functions?
  - 1) If $R_1, R_2, \ldots, R_k$ are submodular, and $a_1, a_2, \ldots a_k > 0$ then $\sum a_i R_i$ is also submodular
  - 2) Natural example:
    - Sets $A_1, A_2, \ldots, A_n$:
    - R(S) = size of union of A_i



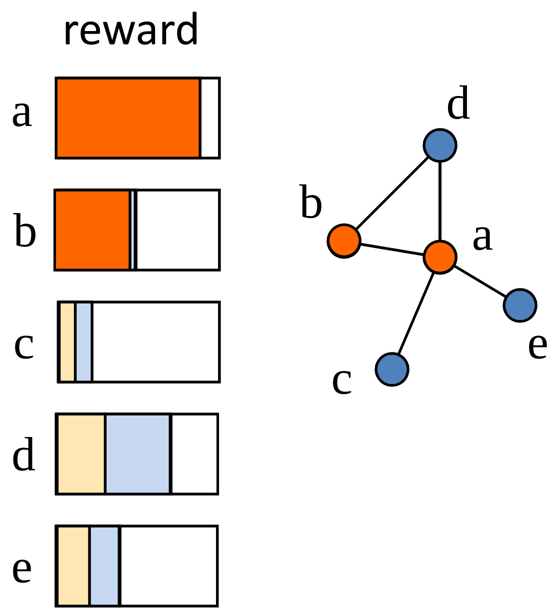
Reward Functions are Submodular

- Objective functions from Battle of Water Sensor Networks competition [Ostfeld et al]:
  - 1) Time to detection (DT)
    - How long does it take to detect a contamination?
  - 2) Detection likelihood (DL)
    - How many contaminations do we detect?
  - 3) Population affected (PA)
    - How many people drank contaminated water?

are all submodular

Background: Submodular functions

Hill-climbing



Add sensor with highest marginal gain

What do we know about optimizing submodular functions?

- A hill-climbing (*i.e.*, greedy) is near optimal (*1-1/e* (~63%) of optimal)
- But
  - 1) this only works for unit cost case (each sensor/location costs the same)
  - 2) Hill-climbing algorithm is slow
    - At each iteration we need to re-evaluate marginal gains
    - It scales as *O(|V|B)*

Towards a New Algorithm

- Possible algorithm: hill-climbing ignoring the cost
  - Repeatedly select sensor with highest marginal gain
  - Ignore sensor cost
- It always prefers more expensive sensor with reward *r* to a cheaper sensor with reward *r-ε*
  → For variable cost it can fail arbitrarily badly
- **Idea**
  - What if we optimize benefit-cost ratio?

$$s_k = \underset{s \in \mathcal{V} \setminus \mathcal{A}_{k-1}}{\operatorname{argmax}} \frac{R(\mathcal{A}_{k-1} \cup \{s\}) - R(\mathcal{A}_{k-1})}{c(s)}$$

Benefit-Cost: More Problems

- Bad news: Optimizing benefit-cost ratio can fail arbitrarily badly
- <u>Example</u>: Given a budget B, consider:
  - 2 locations s_1 and s_2:
    -
    - $)=B$
  - Th
    - $bc(s_1)=2$ and $bc(s_2)=1$
  - So, we first select s_1 and then can not afford s_2
  - → We get reward 2ε instead of B

    Now send ε to 0 and we get arbitrarily bad

What if we take best of both solutions?

Solution: CELF Algorithm

- CELF (cost-effective lazy forward-selection): A two pass greedy algorithm:
  - Set (solution) A: use benefit-cost greedy
  - Set (solution) B: use unit cost greedy
  - Final solution: *argmax(R(A), R(B))*
- How far is CELF from (unknown) optimal solution?
- Theorem: CELF is near optimal
  - CELF achieves $\frac{1}{2}(1-1/e)$ factor approximation
- CELF is much faster than standard hill-climbing

How good is the solution?

- Traditional bound *(1-1/e)* tells us:

  How far from optimal are we even before seeing the data and running the algorithm

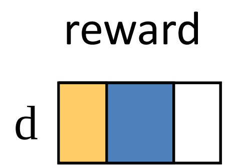
- Can we do better? Yes!

- We develop a new tighter bound. Intuition:
  - Marginal gains are decreasing with the solution size
  - We use this to get tighter bound on the solution

Scaling up CELF algorithm

- Observation:

  Submodularity guarantees that marginal benefits decrease with the solution size

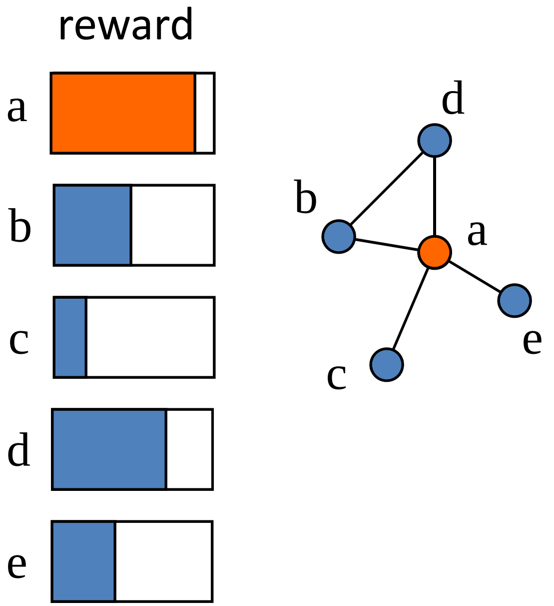


- Idea: exploit submodularity, doing lazy evaluations!

  (considered by Robertazzi et al. for unit cost case)

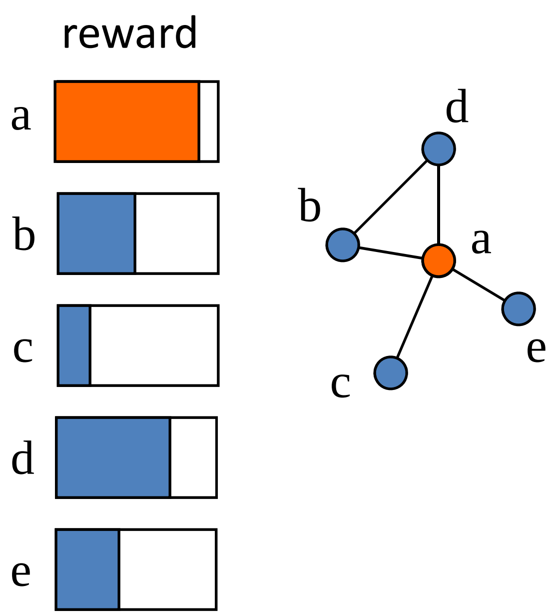
Scaling up CELF

- CELF algorithm – hill-climbing:
  - Keep an ordered list of marginal benefits b_i from previous iteration
  - Re-evaluate b_i only for top sensor
  - Re-sort and prune



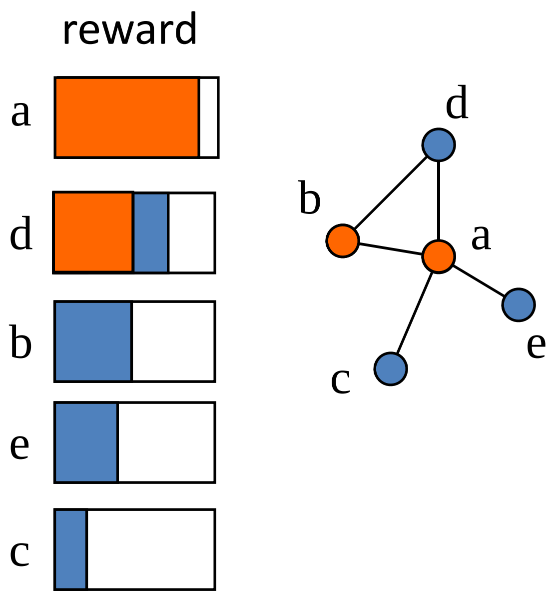
Scaling up CELF

- CELF algorithm – hill-climbing:
  - Keep an ordered list of marginal benefits b_i from previous iteration
  - Re-evaluate b_i only for top sensor
  - Re-sort and prune



Scaling up CELF

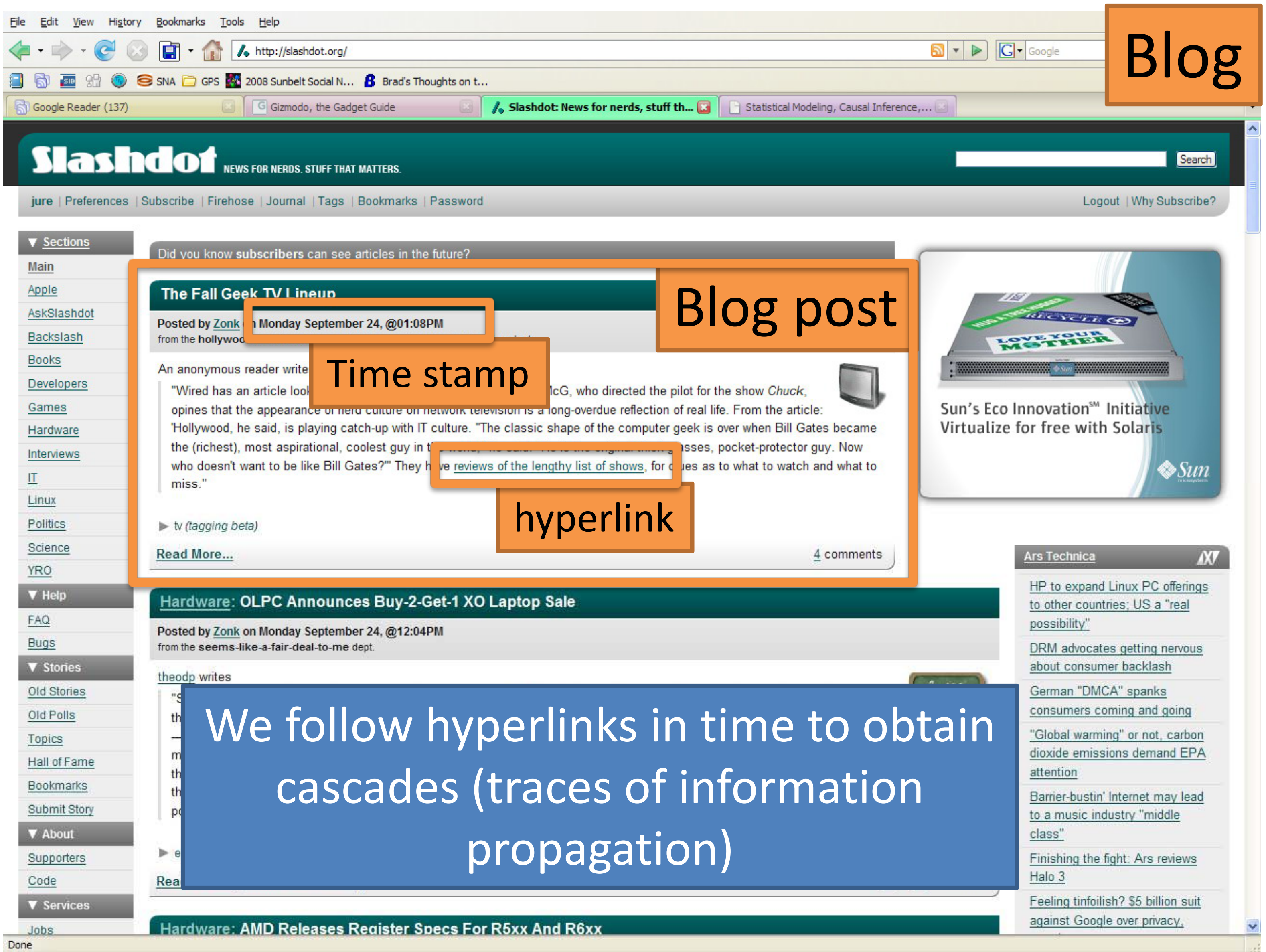
- CELF algorithm – hill-climbing:
  - Keep an ordered list of marginal benefits b_i from previous iteration
  - Re-evaluate b_i only for top sensor
  - Re-sort and prune



Experiments: 2 Case Studies

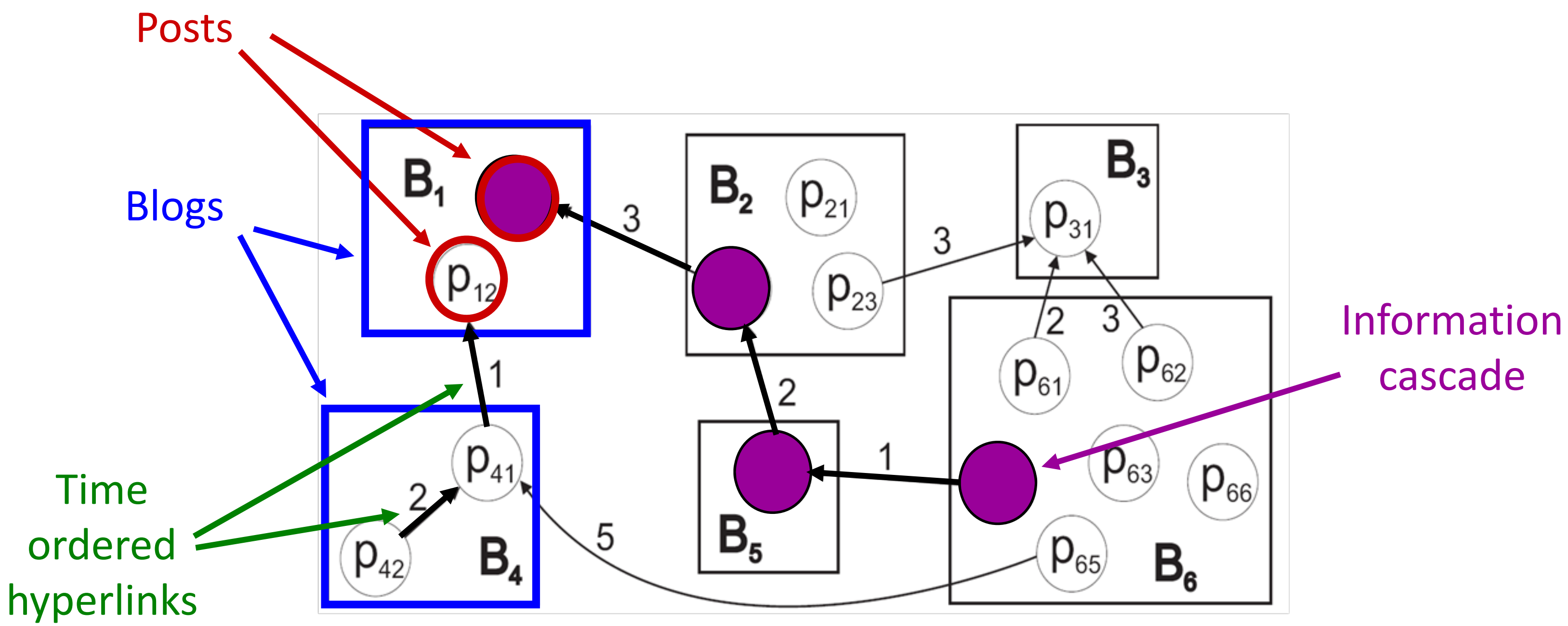
- We have real propagation data
  - Blog network:
    - We crawled blogs for 1 year
    - We identified cascades – temporal propagation of information
  - Water distribution network:
    - Real city water distribution networks
    - Realistic simulator of water consumption provided by US Environmental Protection Agency

Case study 1: Cascades in Blogs



We follow hyperlinks in time to obtain cascades (traces of information propagation)

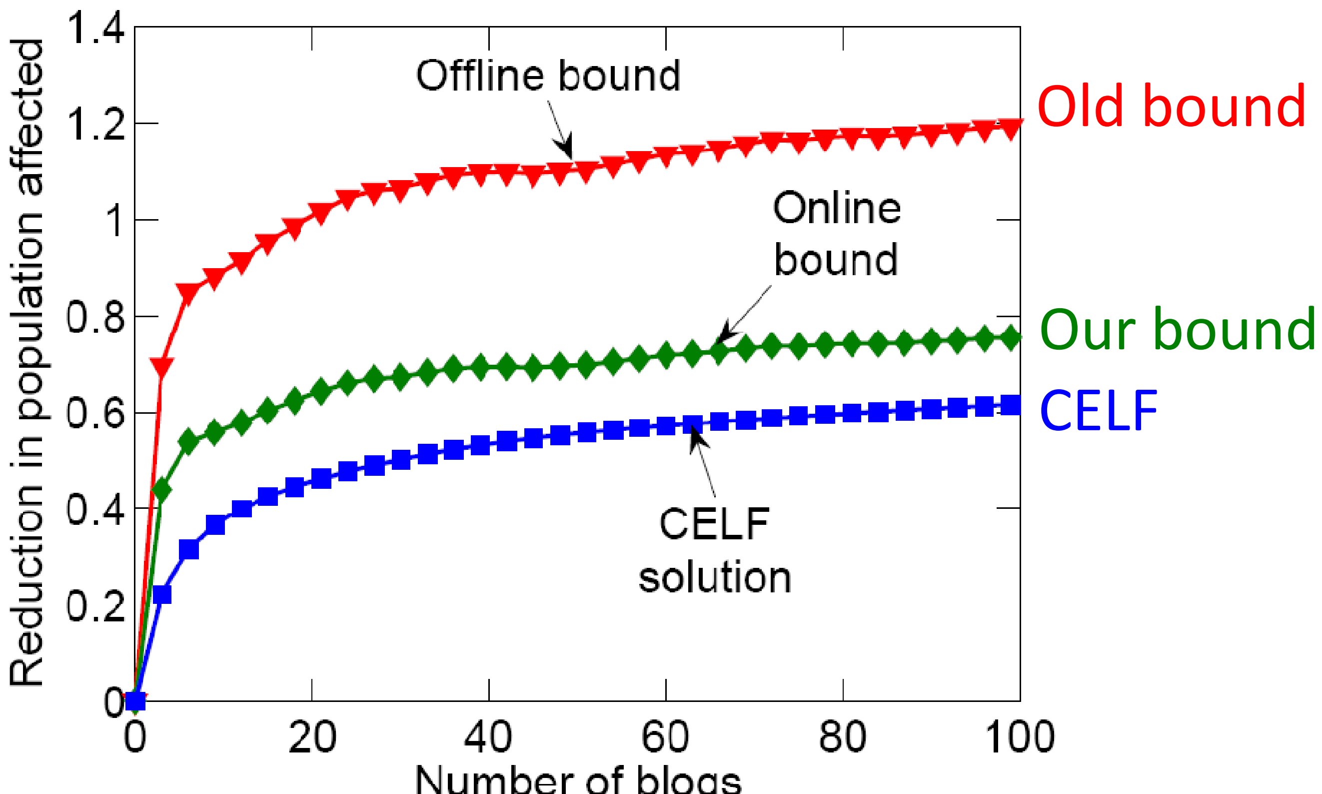
Diffusion in Blogs



- Data – Blogs:
  - We crawled 45,000 blogs for 1 year
  - 10 million posts and 350,000 cascades

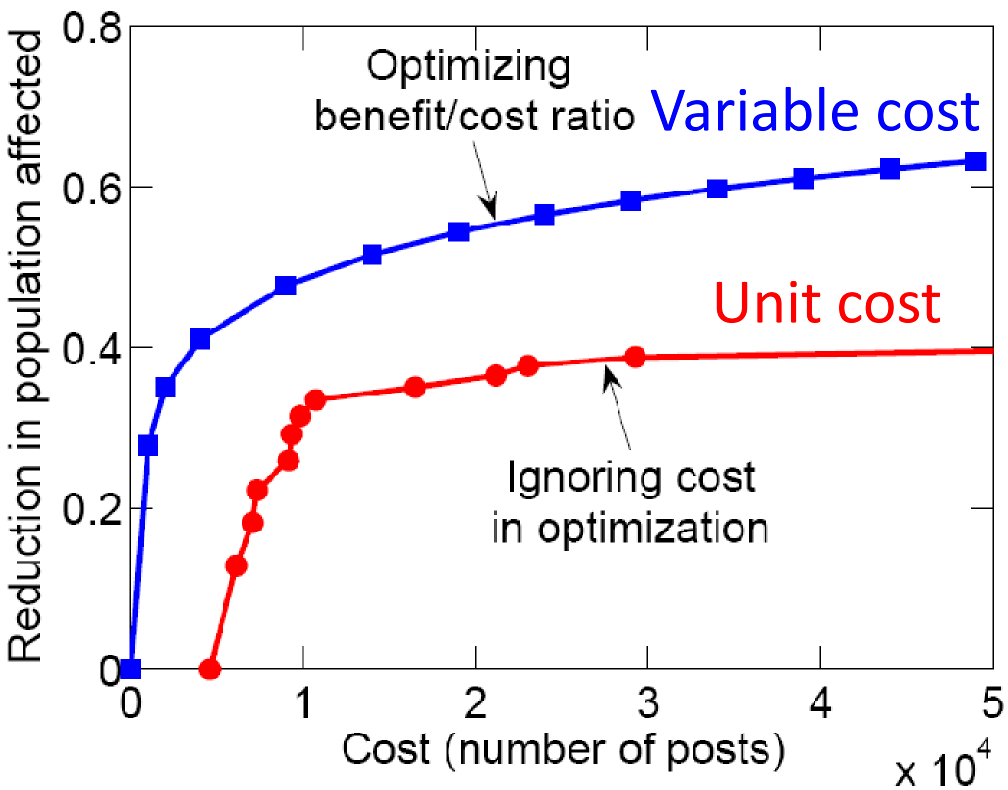
Q1: Blogs: Solution Quality

- Our bound is much tighter
  - 13% instead of 37%



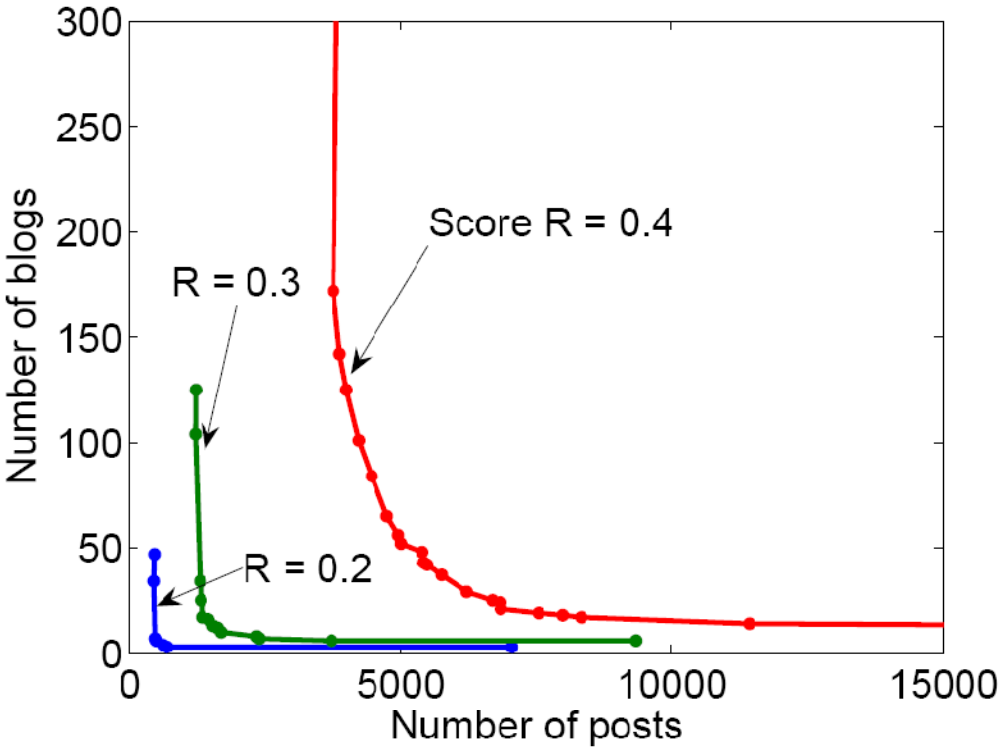
Q2: Blogs: Cost of a Blog

- Unit cost:
  - algorithm picks large popular blogs: `instapundit.com`, `michellemalkin.com`
- Variable cost:
  - proportional to the number of posts
- We can do much better when considering costs



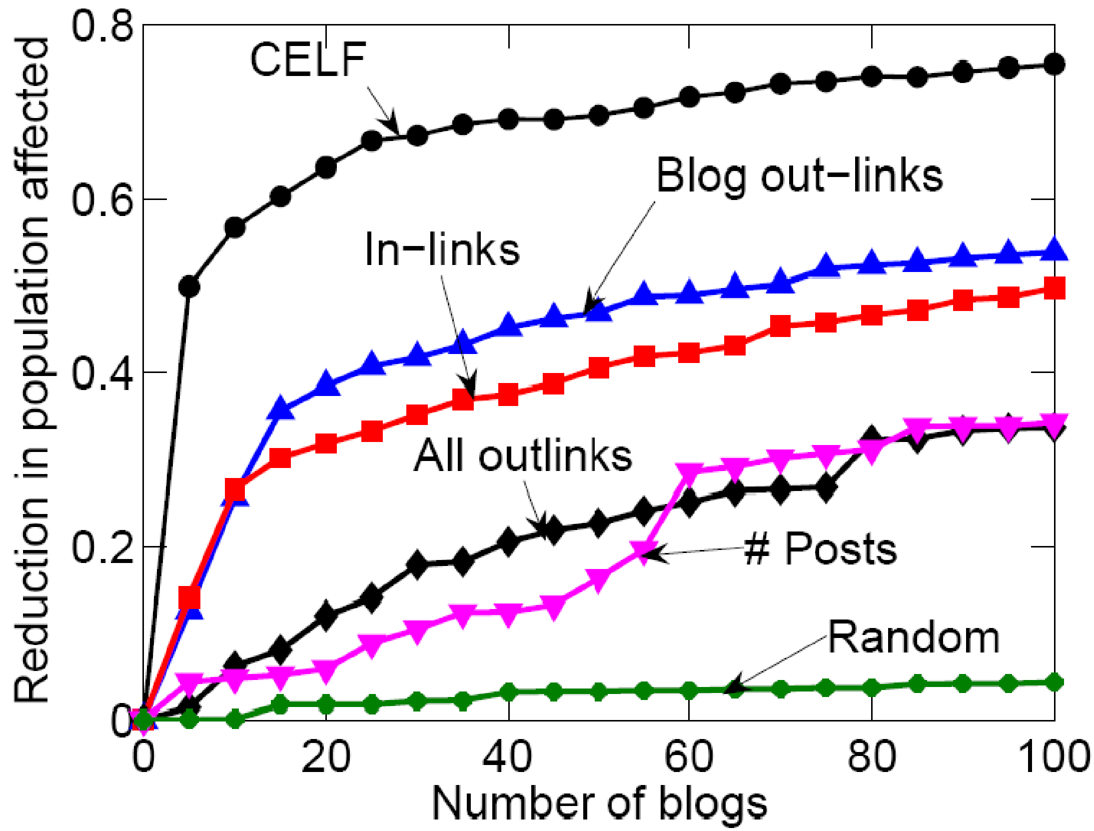
Q2: Blogs: Cost of a Blog

- But then algorithm picks lots of small blogs that participate in few cascades
- We pick best solution that interpolates between the costs
- We can get good solutions with few blogs and few posts



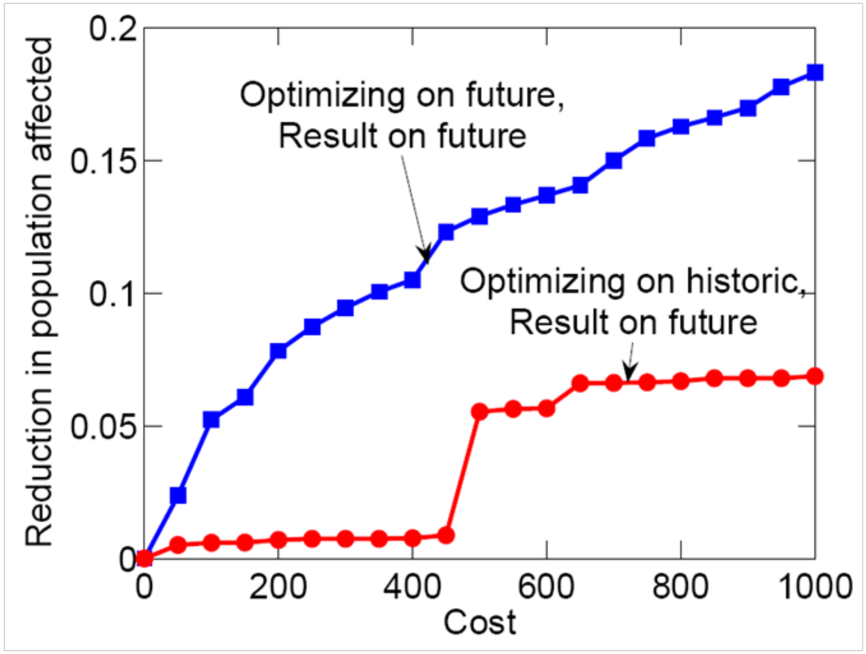
Each curve represents solutions with same final reward

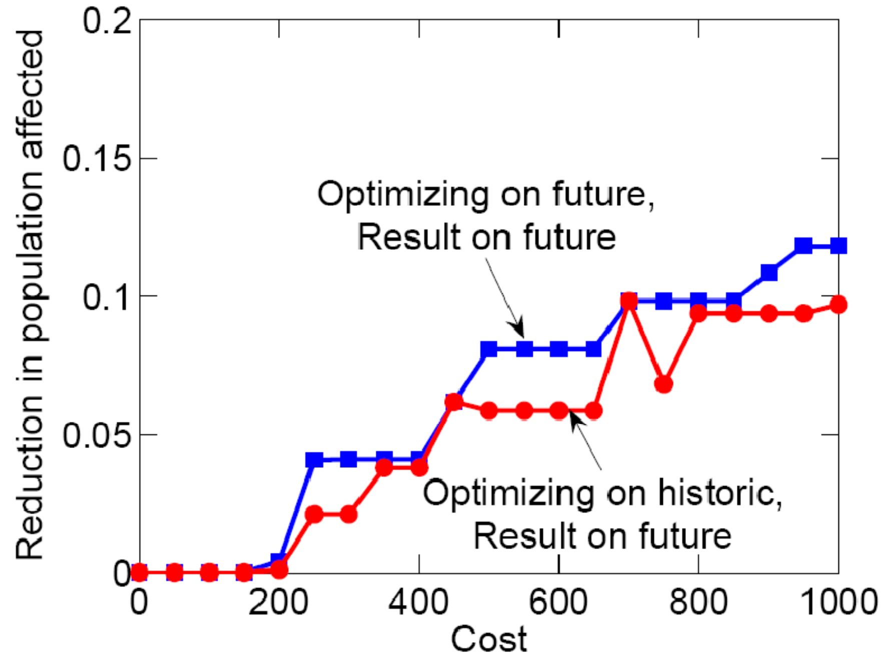
Q4: Blogs: Heuristic Selection



- Heuristics perform much worse
- One really needs to perform optimization

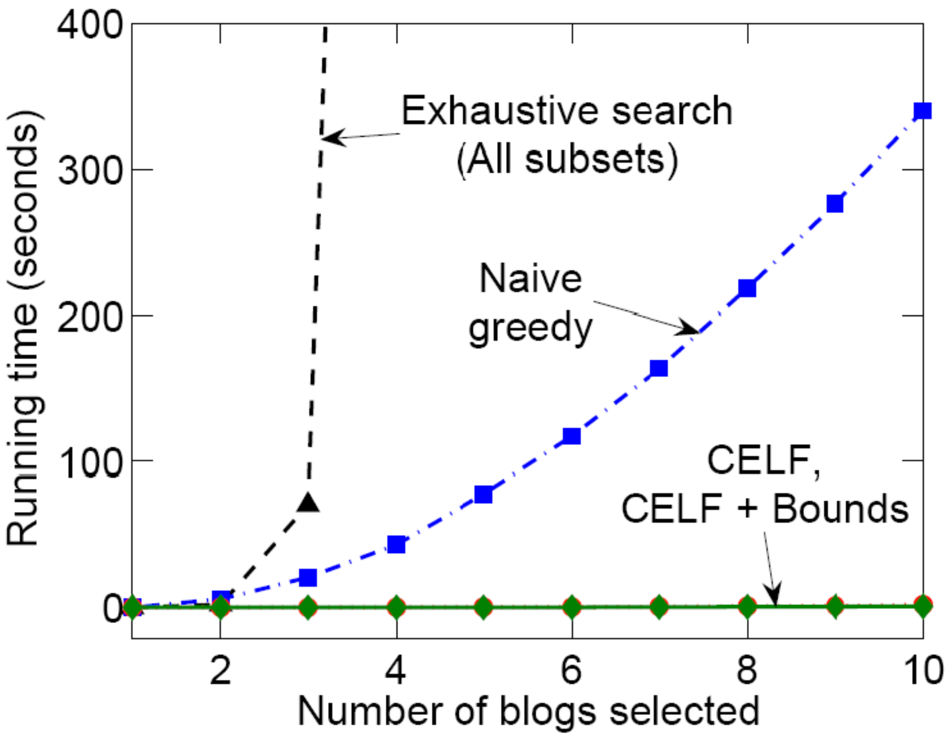
Blogs: Generalization to Future





- We want to generalize well to future (unknown) cascades
- Limiting selection to bigger blogs improves generalization

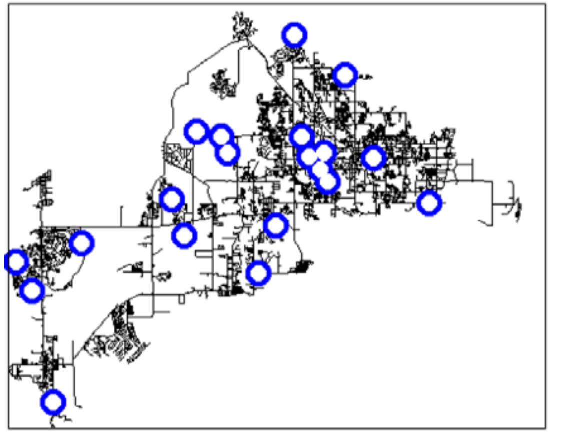
Q5: Blogs: Scalability



- CELF runs **700** times faster than simple hill-climbing algorithm

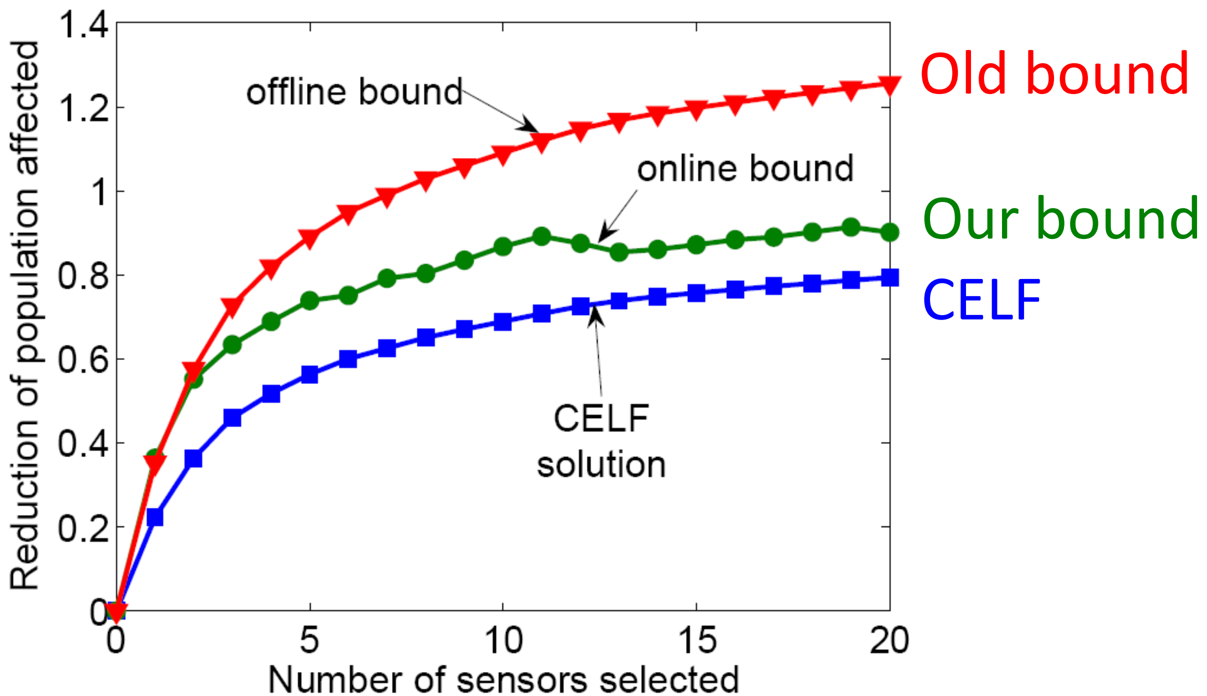
Case study 2: Water Network

- Real metropolitan area water network
  - V = 21,000 nodes
  - E = 25,000 pipes



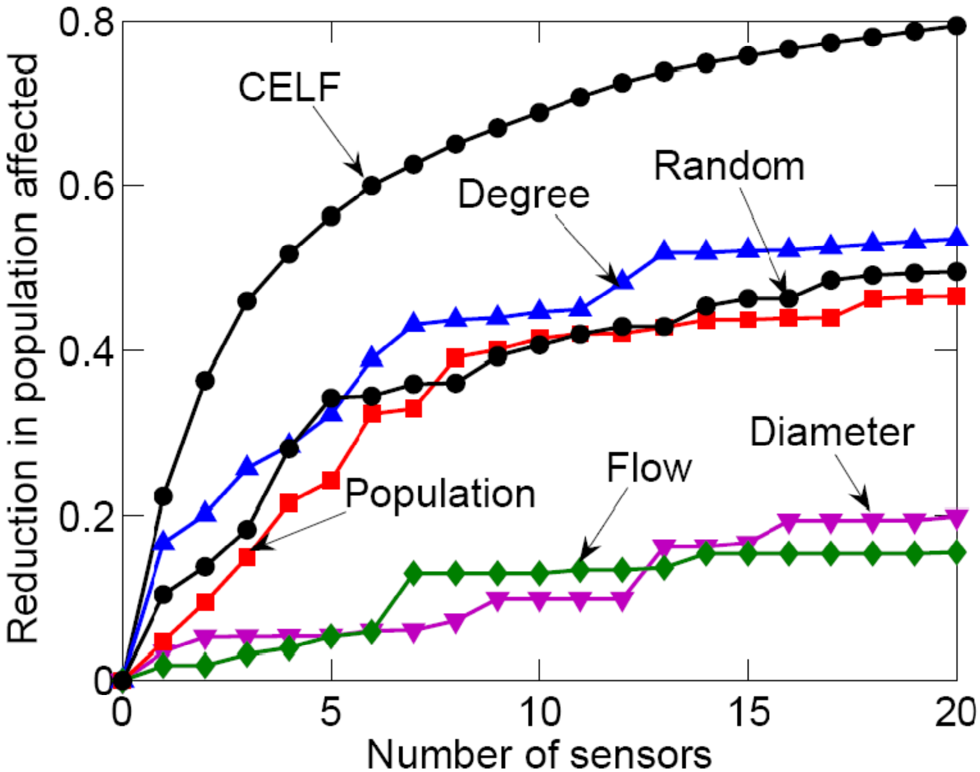
- Use a cluster of 50 machines for a month
- Simulate 3.6 million epidemic scenarios (152 GB of epidemic data)
- By exploiting sparsity we fit it into main memory (16GB)

Water: Solution Quality



- The new bound gives much better estimate of solution quality

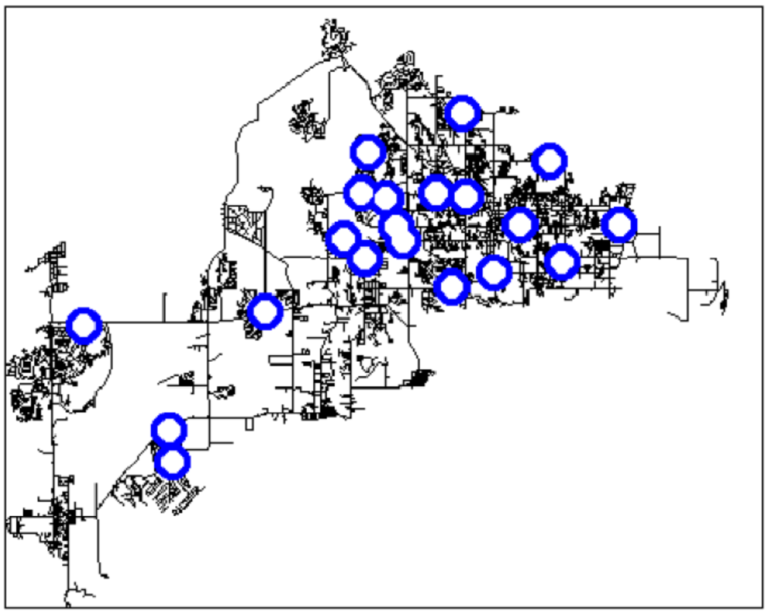
Water: Heuristic Placement



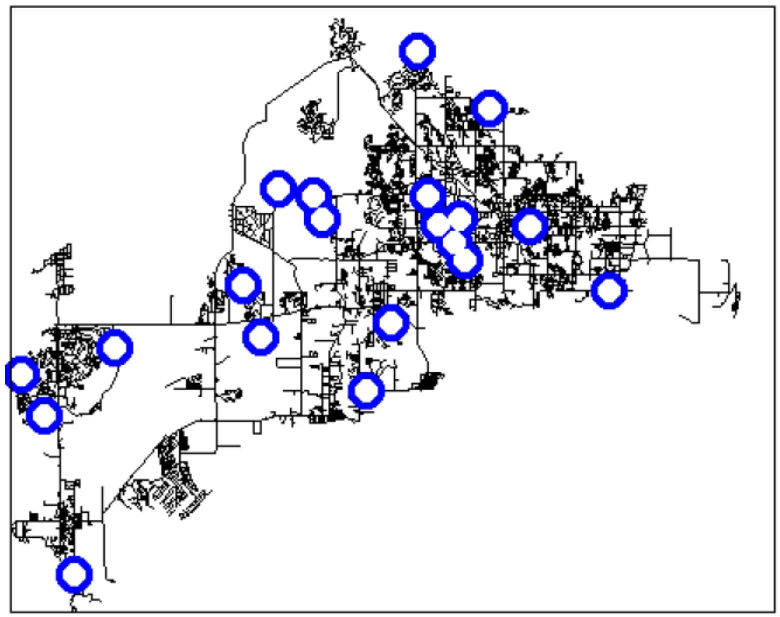
- Heuristics placements perform much worse
- One really needs to consider the spread of epidemics

Water: Placement Visualization

- Different reward functions give different sensor placements

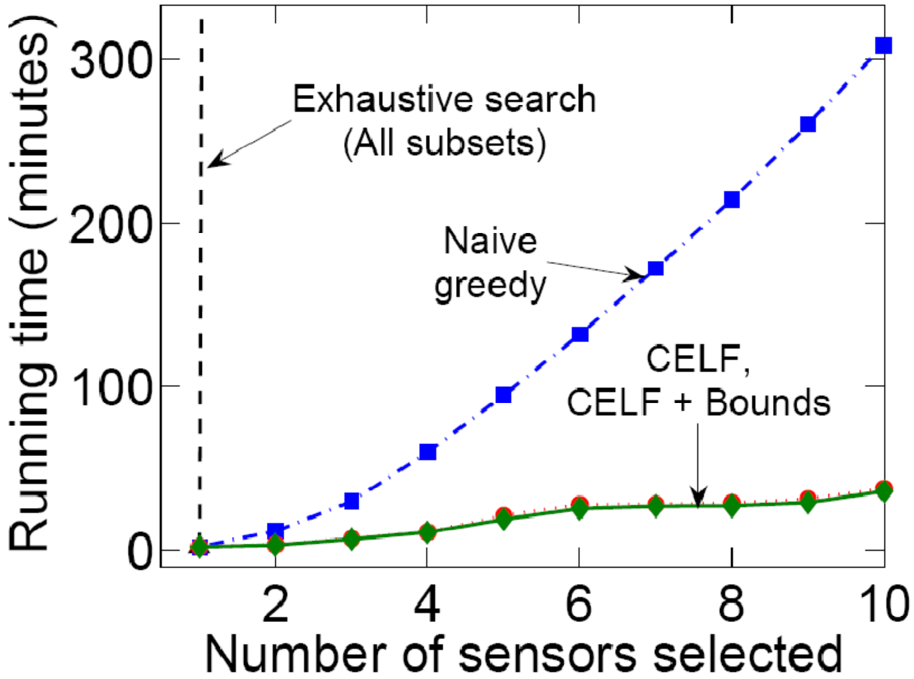


Population affected



Detection likelihood

Water: Algorithm Scalability



- CELF is an order of magnitude faster than hill-climbing

Results of BWSN competition

- *Battle of Water Sensor Networks* competition
- [Ostfeld et al]: count number of non-dominated solutions

Author	#non- dominated (out of 30)
CELF	**26**
Berry et. al.	21
Dorini et. al.	20
Wu and Walski	19
Ostfeld et al	14
Propato et. al.	12
Eliades et. al.	11
Huang et. al.	7
Guan et. al.	4
Ghimire et. al.	3
Trachtman	2
Gueli	2
Preis and Ostfeld	1

Other results

- Many more details:
  - Fractional selection of the blogs
  - Generalization to future unseen cascades
  - Multi-criterion optimization
  - We show that triggering model of Kempe et al is a special case of out setting

Conclusion

- General methodology for selecting nodes to detect outbreaks
- Results:
  - Submodularity observation
  - Variable-cost algorithm with optimality guarantee
  - Tighter bound
  - Significant speed-up (700 times)
- Evaluation on large real datasets (150GB)
  - CELF won consistently

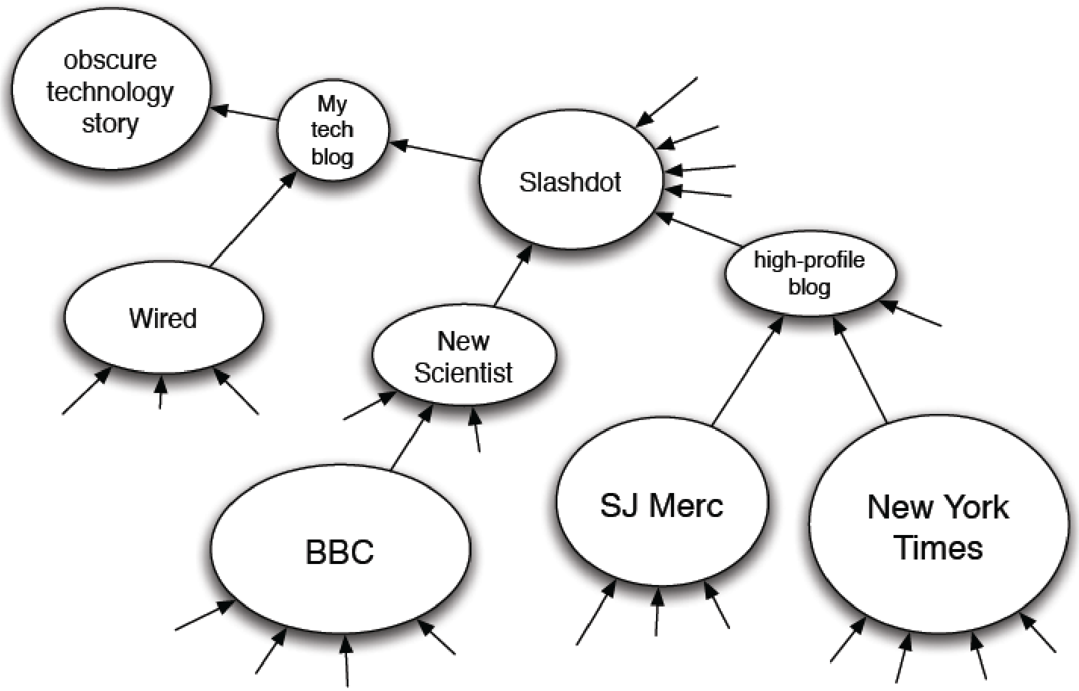
Conclusion and Connections

- Diffusion of Topics
  - How news cascade through on-line networks
  - Do we need new notions of rank?
- Incentives and Diffusion
  - Using diffusion in the design of on-line systems
  - Connections to game theory
- When will one product overtake the other?

Further Connections

- Diffusion of topics [Gruhl et al ‘04, Adar et al ‘04]:
  - News stories cascade through networks of bloggers
  - How do we track stories and rank news sources?
- Recommendation incentive networks [Leskovec-Adamic-Huberman ‘07]:
  - How much reward is needed to make the product “work-of-mouth” success?
- Query incentive networks [Kleinberg-Raghavan ‘05]:
  - Pose a request to neighbors; offer reward for answer
  - Neighbors can pass on request by offering (smaller) reward
  - How much reward is needed to produce an answer?

Topic Diffusion: what blogs to read?



- News and discussion spreads via diffusion:
  - Political cascades are different than technological cascades
- Suggests new ranking measures for blogs

References

- D. Kempe, J. Kleinberg, E. Tardos. Maximizing the Spread of Influence through a Social Network. ACM KDD, 2003.
- Jure Leskovec, Lada Adamic, Bernardo Huberman. The Dynamics of Viral Marketing. ACM TWEB, 2007.
- Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, Matthew Hurst. Cascading Behavior in Large Blog Graphs. SIAM Data Mining, 2007.
- Jure Leskovec, Ajit Singh, Jon Kleinberg. Patterns of Influence in a Recommendation Network. PAKDD, 2006.
- Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, Natalie Glance. Cost-effective Outbreak Detection in Networks. ACM KDD, 2007.