Influence maximisation

Social and Technological Networks

Rik Sarkar

University of Edinburgh, 2019.

Course

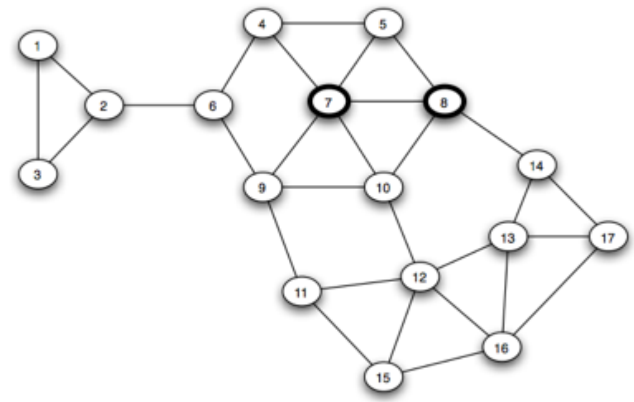
- Piazza forum up at:
  - http://piazza.com/ed.ac.uk/fall2019/infr11124

- Please join. We will post announcements etc there.

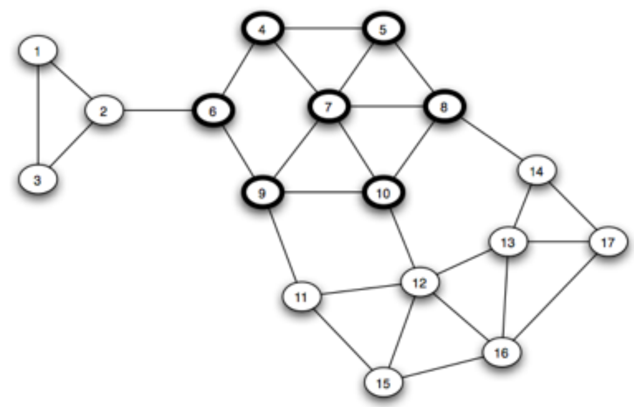
- Its main purpose is as a forum for you to discuss course material
  - Ask questions and answer them. Post relevant things
  - We will answers some questions, not all (and we may be wrong!)
  - Discuss and find answers yourself
  - If you are not sure if your answer is correct, try to articulate the doubt exactly, and the search for answers!

Influence maximisation

- Causing a large spread of cascades
- Viral marketing with limited costs
- Suppose we have a budget to activate k nodes to using our products
- Which k nodes should we activate?



(a) *Two nodes are the initial adopters*



(b) *The process ends after three steps*

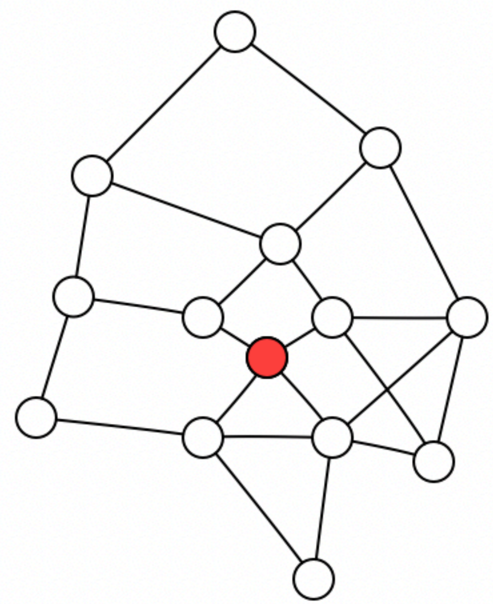
Model of operation

- Suppose each edge e_{uv} has an associated probability p_{uv}
  - Represents strength or closeness of the relation

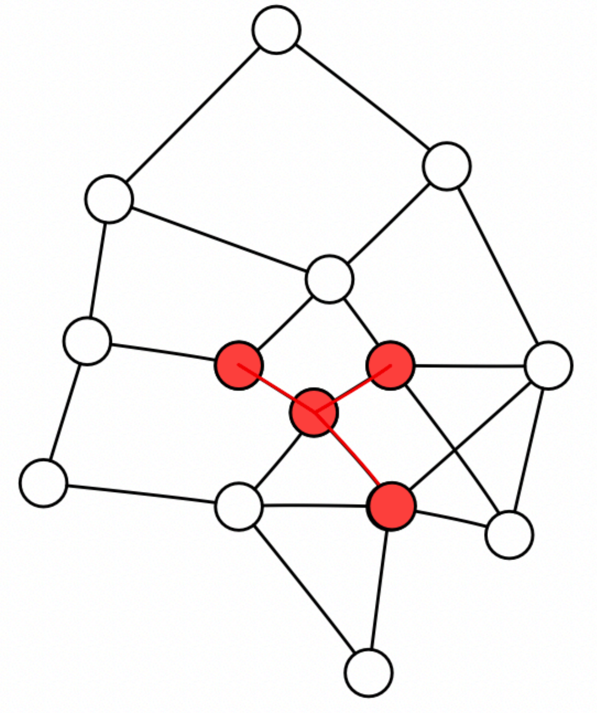
- That is, if u activates, v is likely to pick it up with probability p_{uv}

- Independent activation model

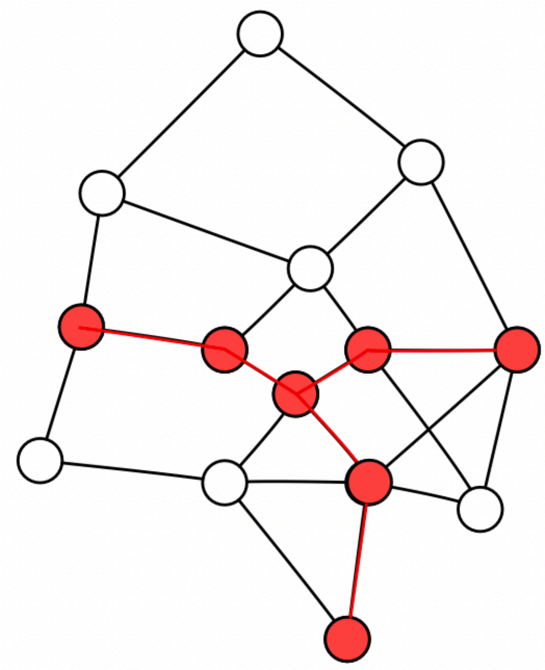
What happens when any one node activates?



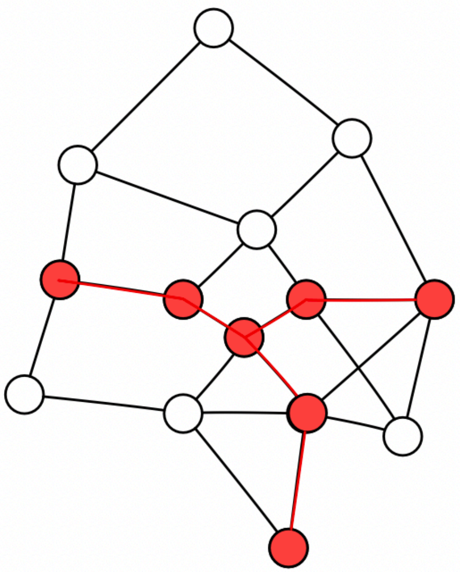
- Some neighbors activate



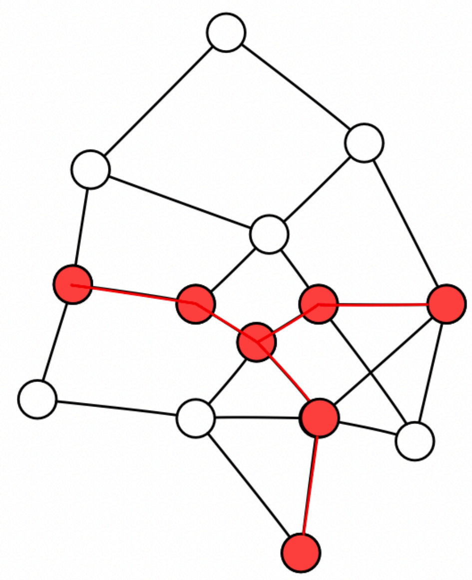
- Some neighbors of neighbors activate …



- The contagion spreads through a connected tree
- Every time we run process, it will activate a random set of nodes starting from the first node
  - It spreads through an edge with the probability for that edge



- For each node v, there is a corresponding activation set S_v
- Question is, which set of k nodes do we want to select so that the union of all S_v is largest



$$\max |\cup S_v|$$

- Naïve strategy
  - Find the activation set for each node
  - Try each possible set of k starting nodes, and pick the best
    - Number of k-sets is $\binom{n}{k}$
  - Second step takes a long time when k is large
  - Better ideas?

- The bad news
- Finding the best possible set of size k is NP-hard
  - Computationally intractable unless *class P = class NP*
  - There is unlikely to be a method much better than the naïve method to find the best set

Approximations

- In many problems, finding the “best” solution is impractical
- In many problems, a “good” solution is quite useful

Approximations

- Usually, the quality of the best solution is written as OPT
- Suppose we find an algorithm produces a result of quality c*OPT
  - It is called a c-approximation
- In case of cascades
  - A c-approximation guarantees reaching at least c*OPT nodes
  - E.g. ½ approximation reaches ½ of OPT nodes

Unknown optimals

- We do not know what OPT is!
- We do not know which set gives OPT

- However, the algorithm we design will guarantee that the result is close to OPT

- For the maximizing activation problem, there is a simple algorithm that gives an approximation of

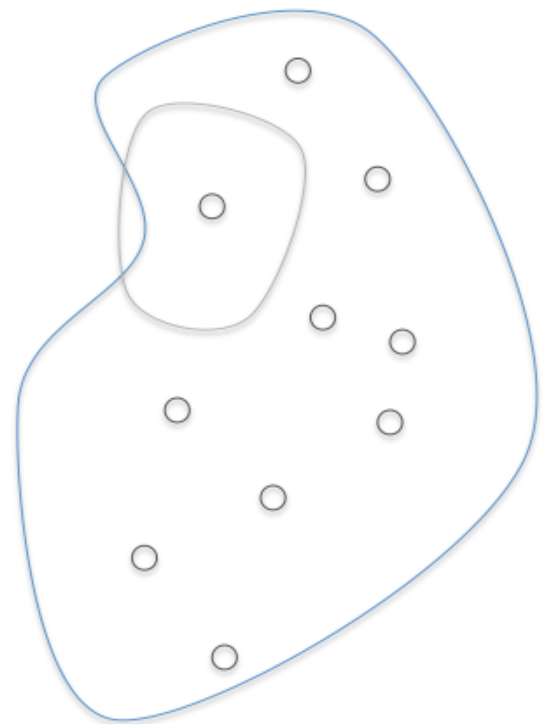
$$\left(1 - \frac{1}{e}\right)$$

- To prove this, we will use a property called *submodularity*
  - A fundamental concept in machine learning

- We will take a diversion to explain submodular maximization through a more intuitive example
- Then come back to cascade or influence maximisation

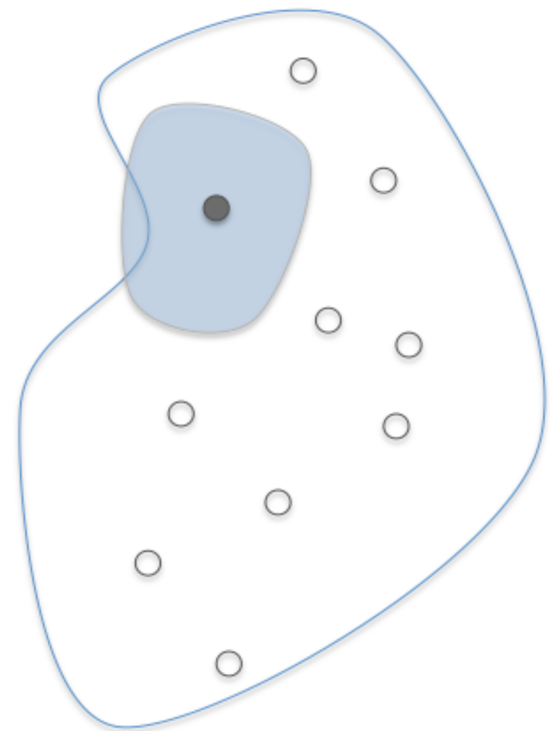
Example: Camera coverage

- Suppose you are placing sensors/cameras to monitor a region (eg. cameras, or chemical sensors etc)
- There are n possible camera locations
- Each camera can “see” a region
- A region that is in the view of one or more sensors is *covered*
- With a budget of k cameras, we want to cover the largest possible area
  - Function f: Area covered



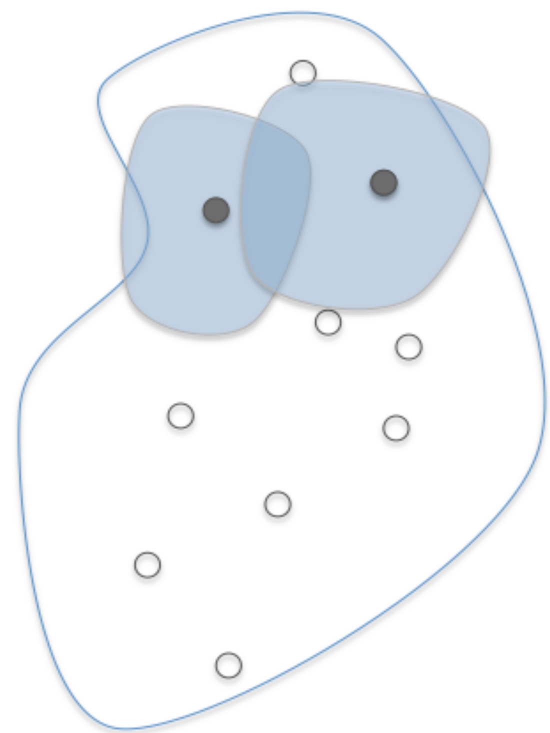
Marginal gains

- Observe:
- Marginal coverage depends on other sensors in the selection



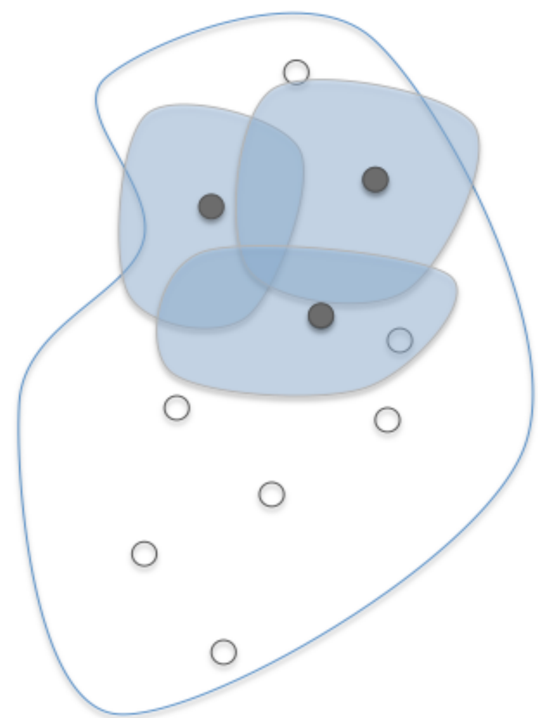
Marginal gains

- Observe:
- Marginal coverage depends on other sensors in the selection



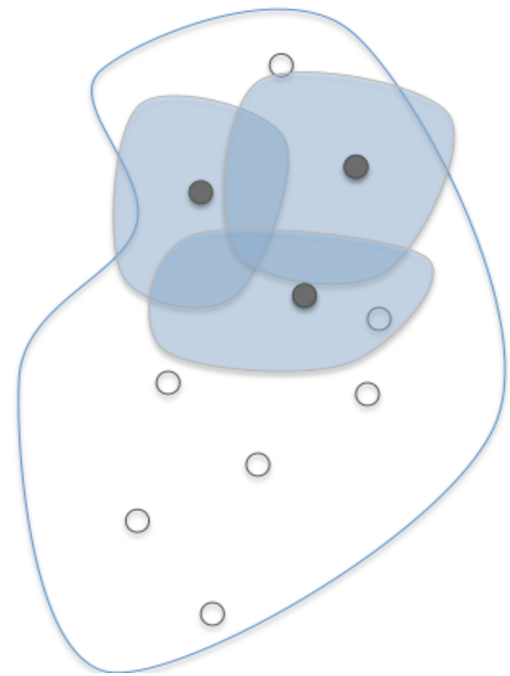
Marginal gains

- Observe:
- Marginal coverage depends on other sensors in the selection
- More selected sensors means less marginal gain from each individual



Submodular functions

- Suppose function f(x) represents the total benefit of selecting x
  - Like area covered
  - And f(S) the benefit of selecting set S
- Function f is submodular if:

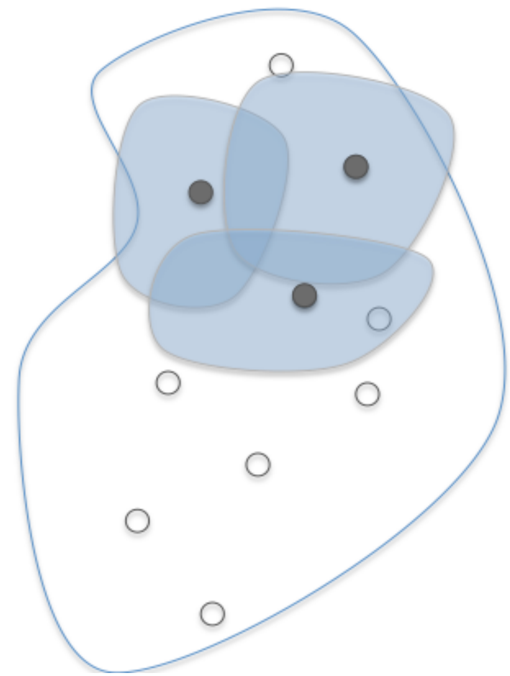


$$S \subseteq T \implies$$

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$

Submodular functions

- Means *diminishing returns*
- A selection of x gives smaller benefits if many other elements have been selected

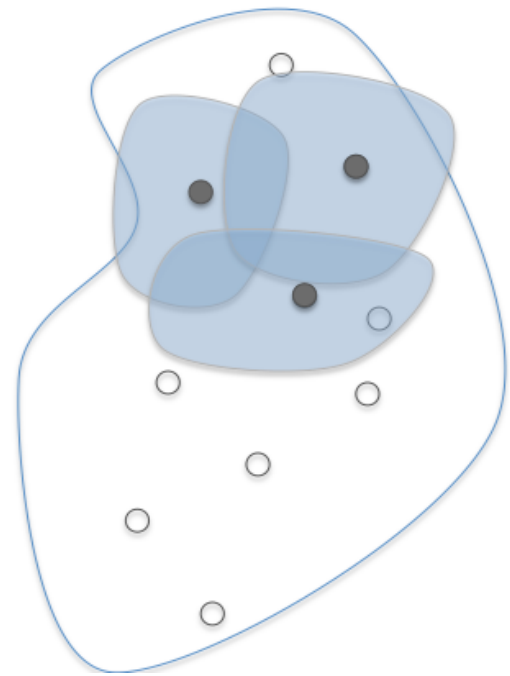


$$S \subseteq T \implies$$

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$

Submodular functions

- Our Problem: select locations set of size k that maximizes coverage
- NP-Hard



$$S \subseteq T \implies$$

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$

Greedy Approximation algorithm

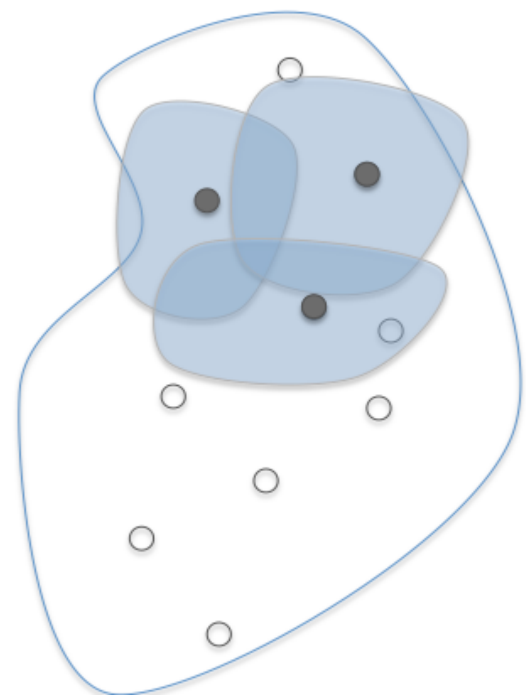
- Start with empty set S = $\varnothing$
- Repeat k times:
- Find v that gives maximum marginal gain:

$$f(S \cup \{v\}) - f(S)$$

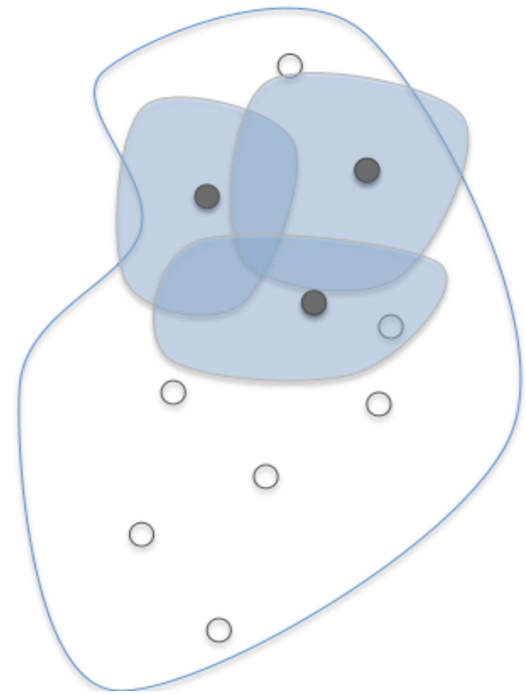
- Insert v into S

- Observation 1: Coverage function is submodular
- Observation 2: Coverage function is monotone:
- Adding more sensors always increases coverage

$$S \subseteq T \Rightarrow f(S) \leq f(T)$$



- This is the same question as influence maximisation
- Which nodes to select, to maximize coverage in a domain



$$S \subseteq T \Rightarrow f(S) \leq f(T)$$

Theorem

- For monotone submodular functions, the greedy algorithm produces a $\left(1 - \frac{1}{e}\right)$ approximation

- That is, the value f(S) of the final set is at least

$$\left(1 - \frac{1}{e}\right) \cdot OPT$$

  – [Nemhauser et al. 1978]

- (Note that this algorithm applies to submodular maximzation problems, not to minimization)

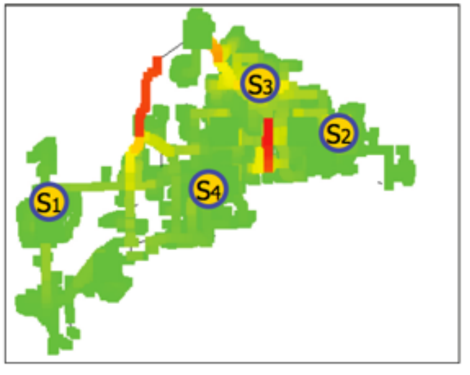
- So, selecting cameras by the greedy algorithm gives a $(1 - 1/e)$ approximation

Applications of submodular optimization

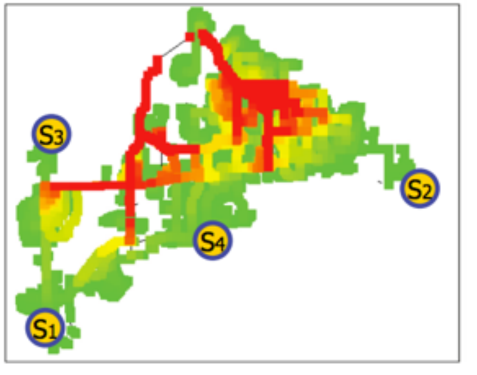
- Sensing the contagion
- Place sensors to detect the spread
- Find "representative elements": Which blogs cover all topics?
- Machine learning selection of sets
- Exemplar based clustering (eg: what are good seed for centers?)
- Image segmentation

Sensing the contagion

- Consider a different problem:
- A water distribution system may get contaminated
- We want to place sensors such that contamination is detected



(c) effective placement



(d) poor placement

Social sensing

- Which blogs should I read? Which twitter accounts should I follow?
  - Catch big breaking stories early
- Detect cascades
  - Detect large cascades
  - Detect them early…
  - With few sensors
- Can be seen as submodular optimization problem:
  - Maximize the "quality" of sensing

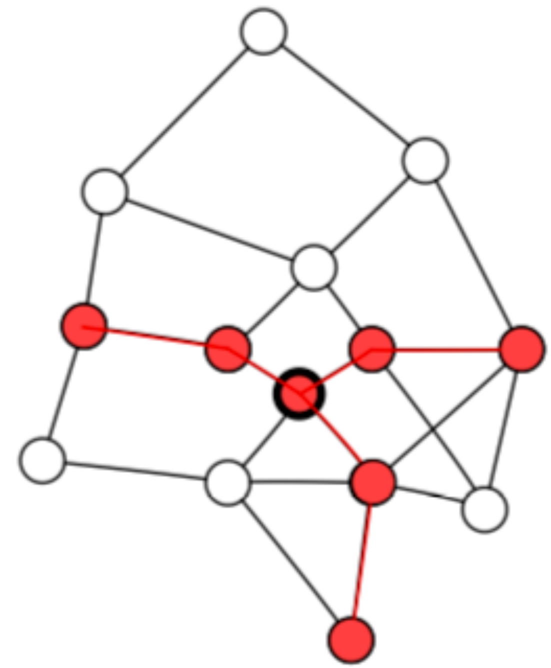
- Ref: Krause, Guestrin; Submodularity and its application in optimized information gathering, TIST 2011

Representative elements

- Take a set of Big data
- Most of these may be redundant and not so useful
- What are some useful “representative elements”?
  - Good enough sample to understand the dataset
  - Cluster representatives
  - Representative images
  - Few blogs that cover main areas…

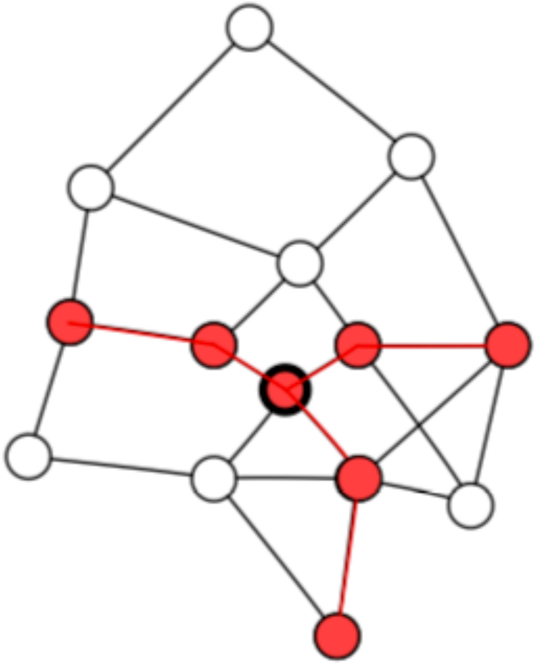
Recap

- Model: Independent activation
  - Contagion propagates along edge e_{uv} with probability p_{uv}
- Choose set of k starting nodes to get max coverage



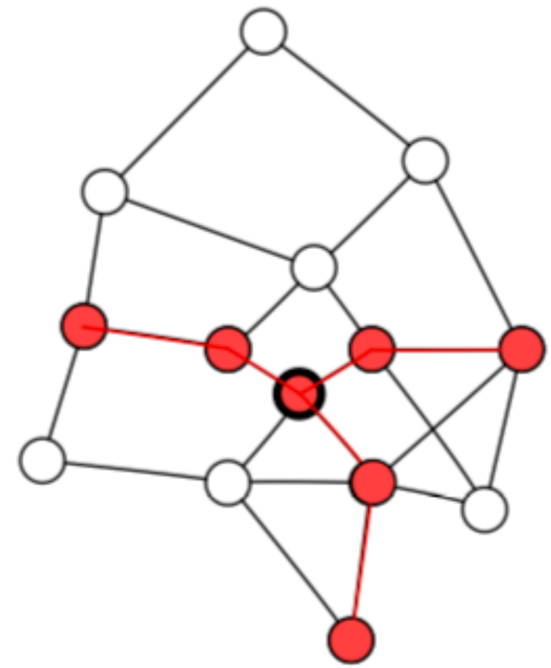
Recap

- Suppose we magically know each activation set S_v that will be infected starting at node v
  - Let us call this behavior X_1
- Finding the best set of k nodes (or equivalently sets S) is hard
- We are looking for approximation



Recap

- Greedy algorithm:
  - Selecting the set S_v of max marginal coverage

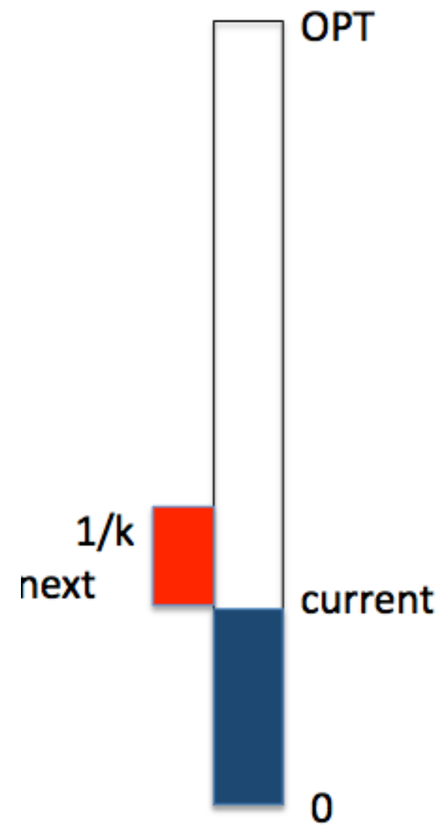


- Gives approximation

$$\left(1 - \frac{1}{e}\right) \cdot OPT$$

Proof

- Idea:
- OPT is the max possible
- At every step there is at least one element that covers at least 1/k of remaining:
  - So ≥ (OPT - current) * 1/k
- Greedy selects one such element

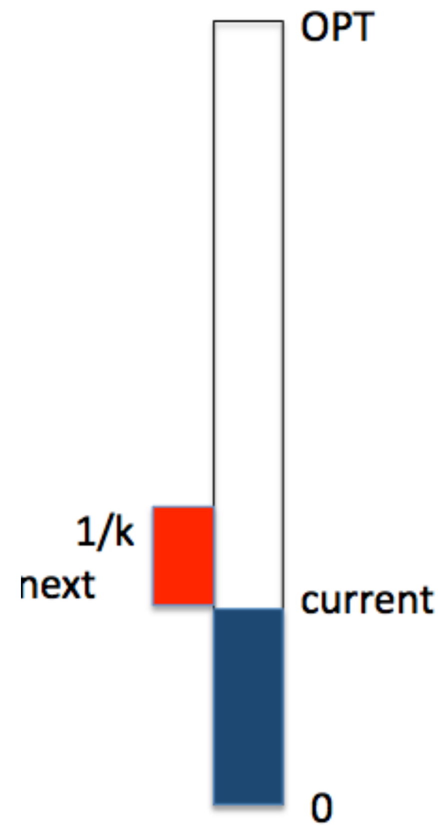


Proof

- Idea:
- At each step coverage remaining becomes

$$\left(1 - \frac{1}{k}\right)$$

- Of what was remaining after previous step



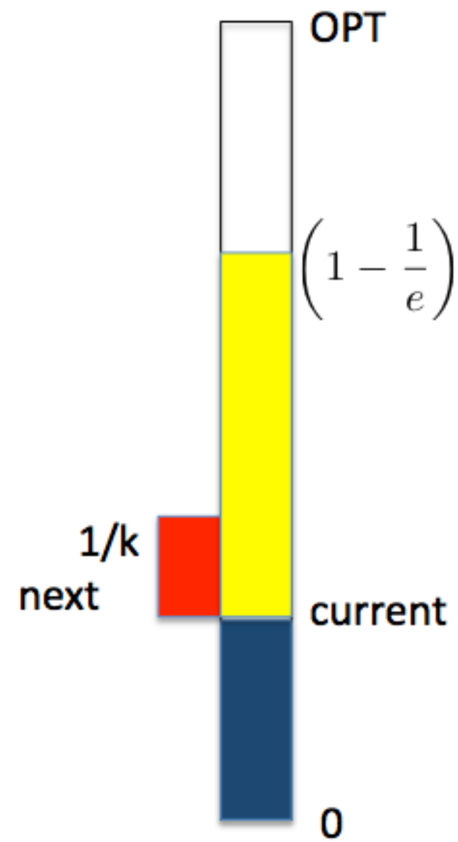
Proof

- After k steps, we have remaining coverage of OPT

$$\left(1 - \frac{1}{k}\right)^k \simeq \frac{1}{e}$$

- Fraction of OPT covered:

$$\left(1 - \frac{1}{e}\right)$$



Proof of the main claim

- At every step there is at least one element that covers at least 1/k of remaining
- Suppose the unknown set of elements that gives OPT is given by set C, so OPT = f(C)

- And suppose S_i is the set selected by greedy upto step i

- Claim: At every step there is at least one element in $C - S_i$ that covers 1/k of remaining: $(f(C) - f(S_i)) * 1/k$

Proof of the main claim

- At every step there is at least one element that covers 1/k of remaining: $(f(C) - f(S_i)) * 1/k$
- At step 0: Suppose to the contrary, there is no such element.
  - Then C cannot give OPT: contradiction.
  - So there is at least one such element

Proof of the main claim

- At any step S_i,
  - We can add all k elements from C to get at least OPT
  - So, at least 1 element of C gives $(f(C) - f(S_i)) * 1/k$
- Now consider Greedy
  - If greedy chose s_i at step i, that is because it gives at least as much marginal gain as any element in C
    - So, s_i covers at least $(f(C) - f(S_i))/k$

Homework

- Write out the proof nicely!

- Given a known behavior X_1 (we know activation sets S_v)
  - Greedy algorithm gives approximation
- But our model is probabilistic
- Each possible behavior X_i occurs with some probability p_i
- We have to prove that the expected behavior in the model is submodular, and therefore can use a greedy algorithm

- Theorem:
  - Positive linear combinations of monotone submodular functions is monotone submodular

- We sum over all possible X_i, weighted by their probability p_i.

- Non-negative linear combinations of submodular functions are submodular,
  - Therefore the sum of all X is submodular
  - (homework!)

Linear threshold model

- Linear contagion threshold model:
- Also submodular and monotone
- Proof ommitted.
  - If you are interested, see additional reading: Kempe, Kleinberg, Tardos; KDD03

The algorithm

- Estimate behaviours X_i and associated p_i
  - Through repeated simulations
  - Current topic of research
- Use greedy algorithm to maximise expected marginal gains

Observation on how the result is approached

- Topic & motivation:
  - Social networks, advertising, adoption etc
- Model
  - Independent activation
    - Assume we are given a graph. For each edge uv we have a probability p_{uv} of transmitting contagion etc
- Problem statement
  - Define influence maximisation: Maximise the number of nodes activated
  - Starting with at most k nodes.

- Result: Constant factor (1 – 1/e) approximation algorithm.

- Homework: write this out formally.

Problem with submodular maximization

- Can be expensive!
- Each iteration costs O(n): have to check each element to find the best
  - May be more: “checks” are complex and depend on current selection
- Problem in large datasets
- Distributed cluster computation can help
  - Split data into multiple computers
  - Compute and merge back results: Works for many types of problems

- Ref: Mirzasoleiman, Karbasi, Sarkar, Krause; Distributed submodular maximization: Finding representative elements in massive data. NIPS 2013.

Summary

- Approximation algorithms
- Critical in practical scenario, since "perfect" answer may be elusive
  - We can find approximations without even knowing the OPT!
- Critical in Machine learning
  - Learning is always approximate
  - We never know the perfect answer for future
  - Learning theory relies on probability and approximations
- Submodular optimisations are a powerful set of tools