Sparsification of Influence Networks

Michael Mathioudakis[1], Francesco Bonchi[2], Carlos Castillo[2], Aris Gionis[2], Antti Ukkonen[2]

[1]University of Toronto, Canada

[2]Yahoo! Research Barcelona, Spain

Introduction

online social networks

facebook 750m users

twitter 100m+ users

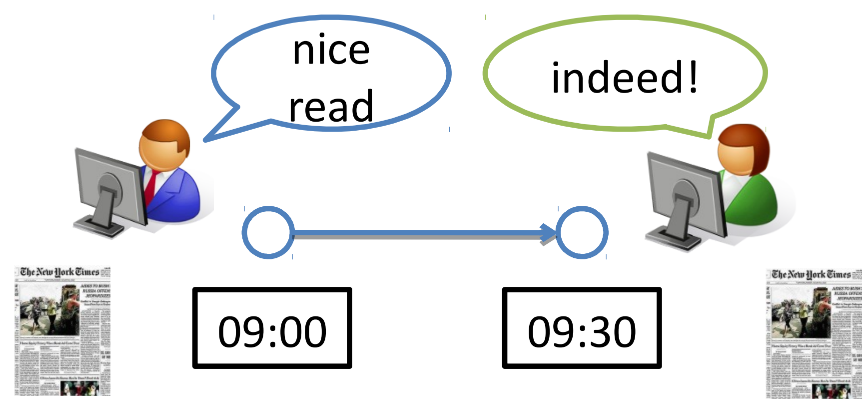
users perform actions

post messages, pictures, videos

connected with other users

interact, influence each other

actions propagate



Problem

which connections are most important
for the propagation of actions?

sparsify network
eliminate large number of connections
keep important connections

sparsification: a data reduction operation
network visualization
efficient graph analysis

What We Do

technical framework

sparsify network according to observed activity

keep connections that best explain propagations

our approach

social network & observed propagations

learn independent cascade model (ICM)

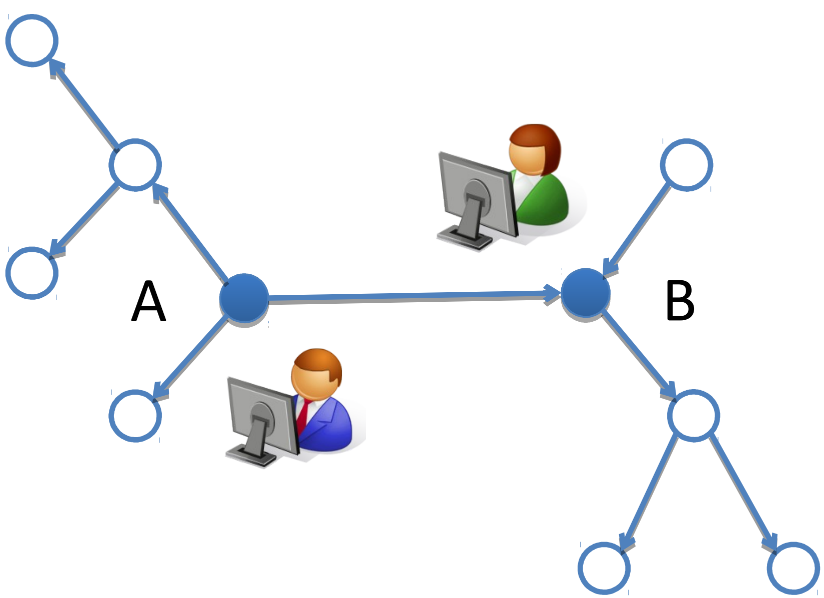
select **k** connections

most likely to have produced propagations

Outline

- introduction
- setting
  - social network
  - propagation model
- sparsification
  - optimal algorithm
  - greedy algorithm: spine
- experiments

Social Network



users – nodes

B follows A – arc A→B

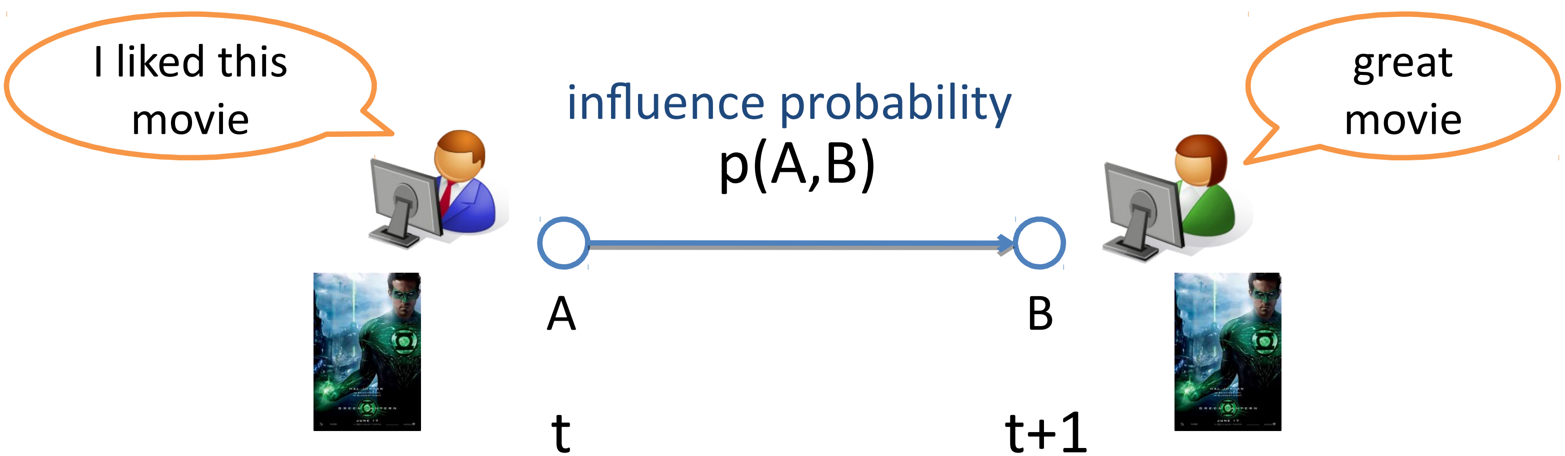
Propagation of Actions

users perform actions

actions propagate

independent **c**ascade **m**odel

propagation of an action unfolds in timesteps

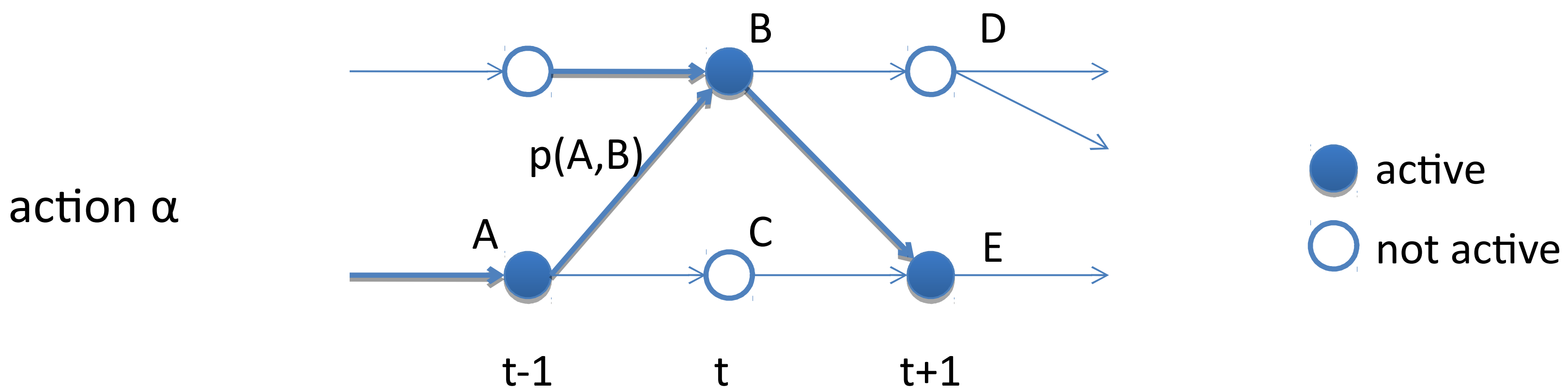


Propagation of Actions

icm generates propagations

sequence of activations

likelihood



Estimating Influence Probabilities

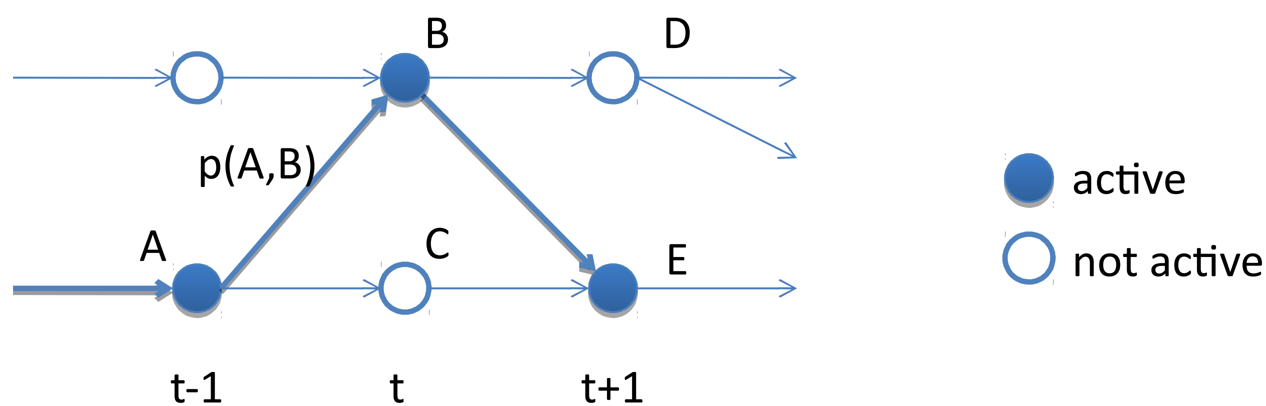
social network
+
set of
propagations

max likelihood

EM – [Saito et.al.]

p(A,B)

action α



Outline

- introduction
- setting
  - social network
  - propagation model
- sparsification
  - optimal algorithm
  - greedy algorithm: spine
- experiments

Sparsification

social network

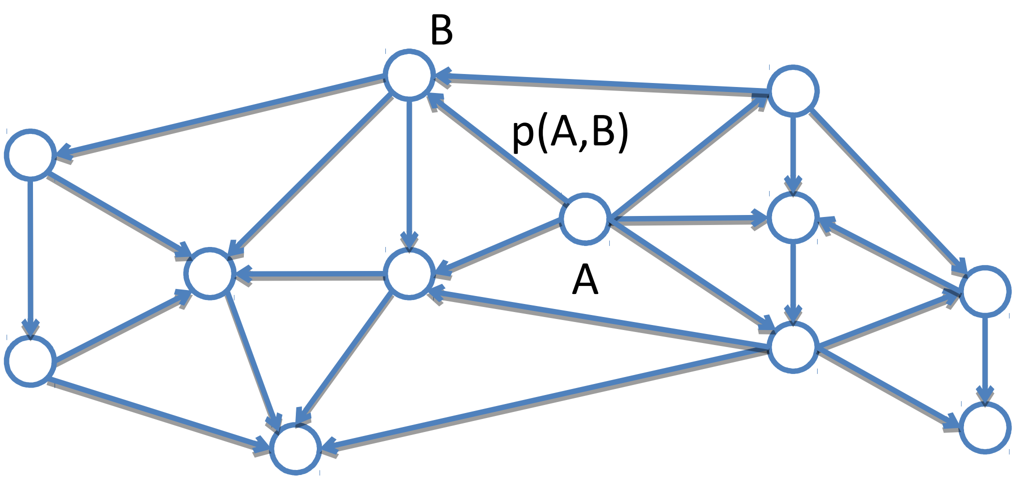
p(A,B)

set of propagations

→

k arcs

most likely to explain all propagations



Sparsification

social network

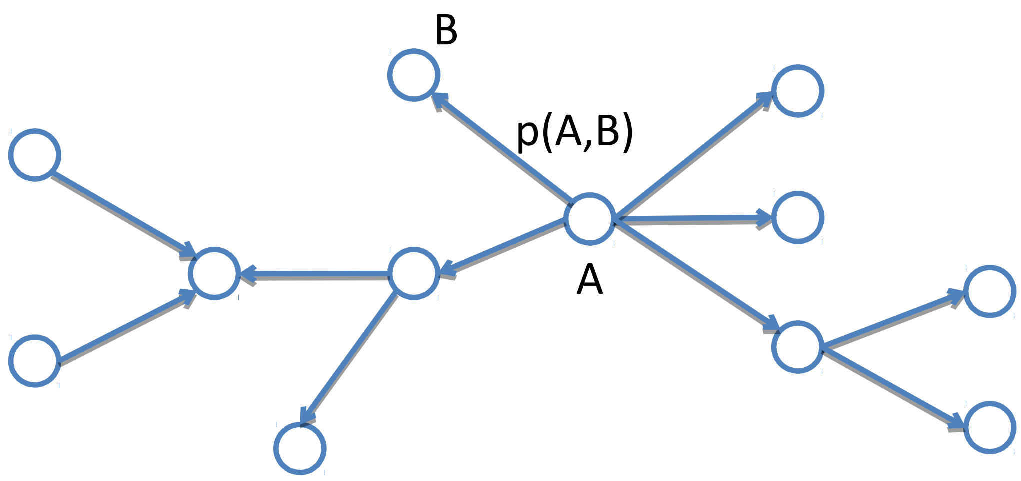
p(A,B)

set of propagations

→

k arcs

most likely to explain all propagations



Sparsification

not the k arcs with largest probabilities

NP-hard and inapproximable

difficult to find solution with non-zero likelihood

How to Solve?

brute-force approach

try all subsets of k arcs?

no

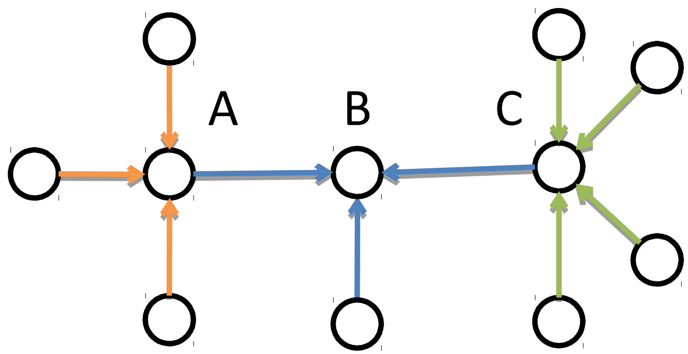
break down into smaller problems

combine solutions

Optimal Algorithm

sparsify separately incoming arcs of individual nodes

optimize corresponding likelihood



$k_A + k_B + k_C = k$

dynamic programming

optimal solution

however…

Spine

sparsification of **i**nfluence **ne**tworks

greedy algorithm

efficient, good results

two phases

phase 1

try to obtain a non-zero-likelihood solution

$k_0 < k$ arcs

phase 2

build on top of phase 1

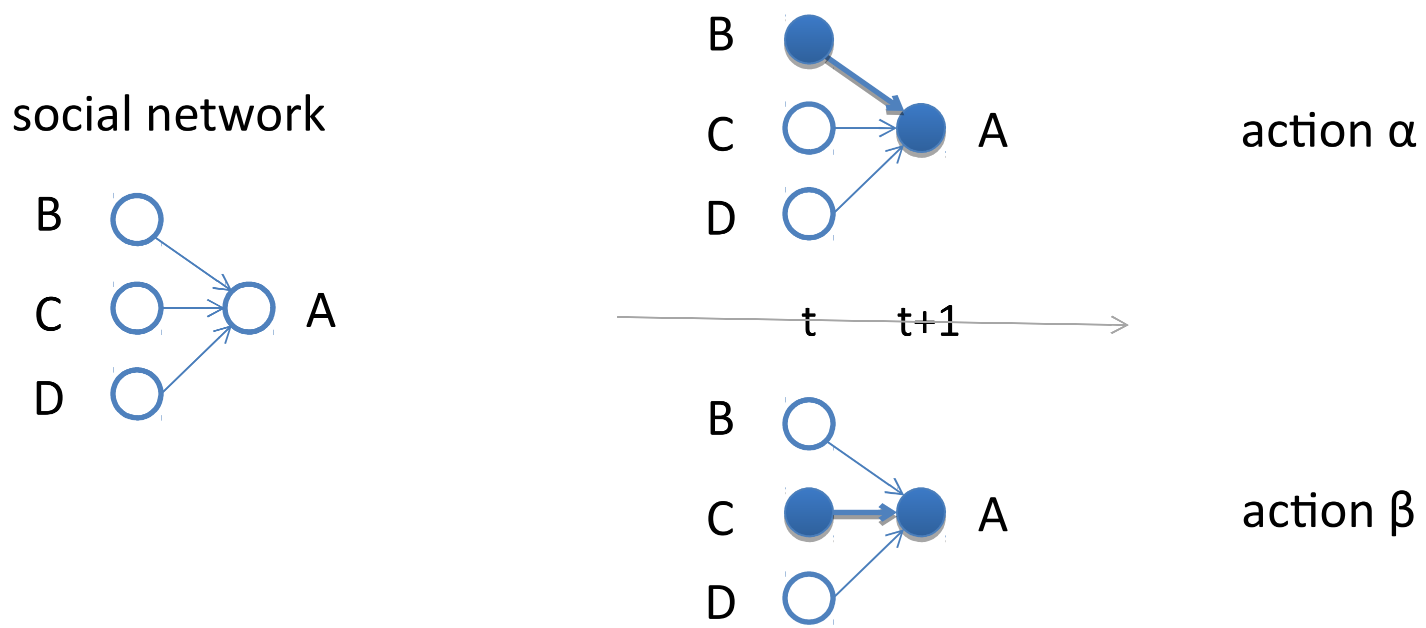
Spine – Phase 1

phase 1

obtain a non-zero-likelihood solution

select greedily arcs that participate in most propagations

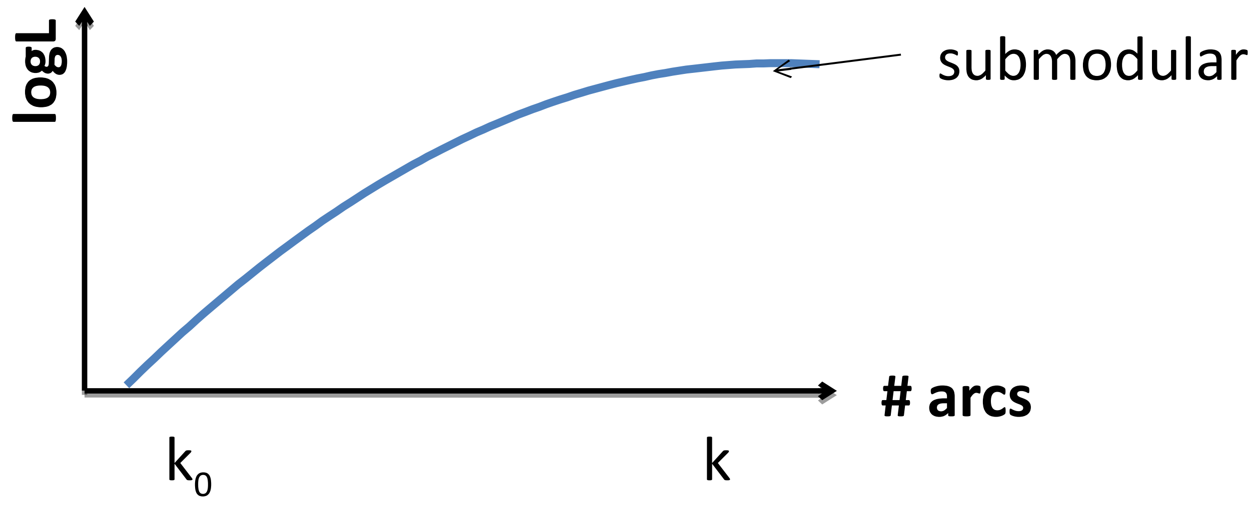
until all propagations are explained



Spine – Phase 2

add one arc at a time, the one that offers

largest increase in likelihood



approximation guarantee
for phase 2

Outline

- introduction
- setting
  - social network
  - propagation model
- sparsification
  - optimal algorithm
  - greedy algorithm: spine
- experiments

Experiments

datasets

meme.yahoo.com

actions: postings (photos), nodes: users, arcs: who follows whom

data from 2010

memetracker.org

actions: mentions of a phrase, nodes: blogs & news sources, arcs: who links to whom

data from 2009

Experiments

sampled datasets of different sizes

Dataset	Actions	Arcs	Arcs, prob > 0
YMeme-L	26k	1.25M	430k
YMeme-M	13k	1.15M	380k
YMeme-S	5k	466k	73k
MTrack-L	9k	200k	7.8k
MTrack-M	120	110k	1.4k
MTrack-S	780	78k	768

YMeme meme.yahoo.com
MTrack memetracker.org

Experiments

algorithms

optimal

(very inefficient)

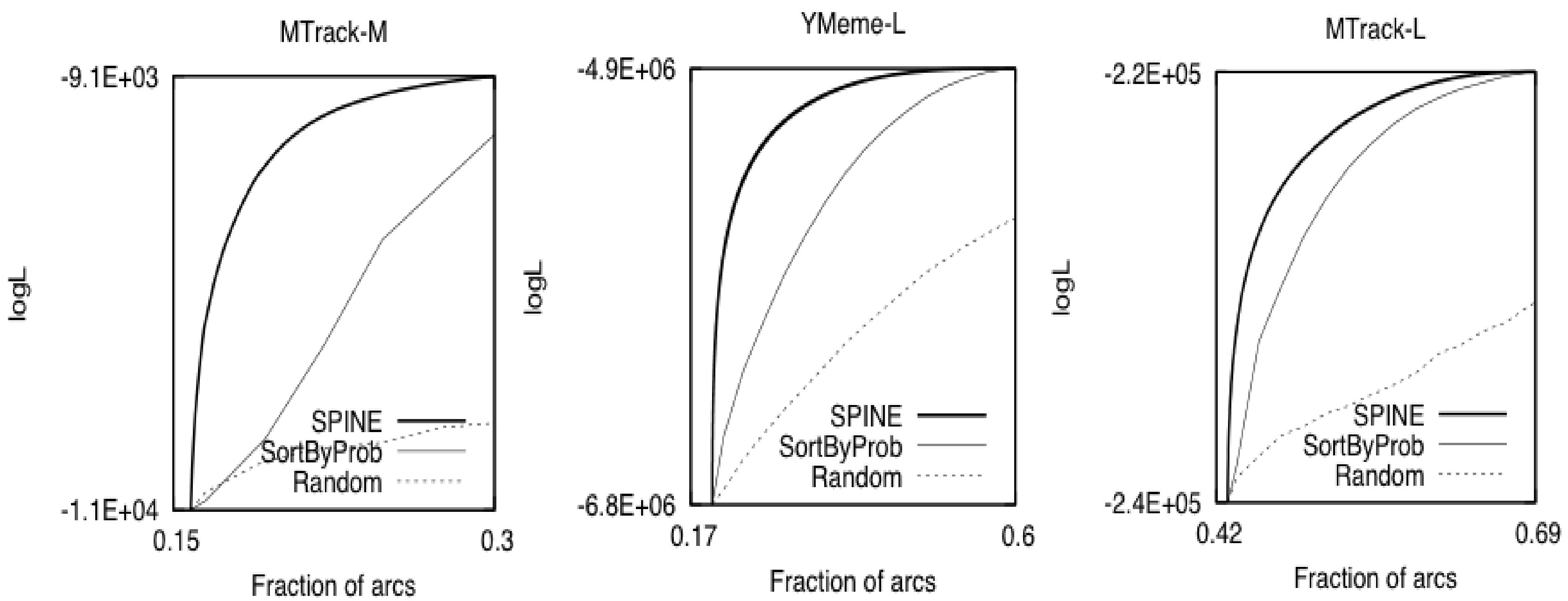
spine

(a few seconds to 3.5hrs)

by arc probability

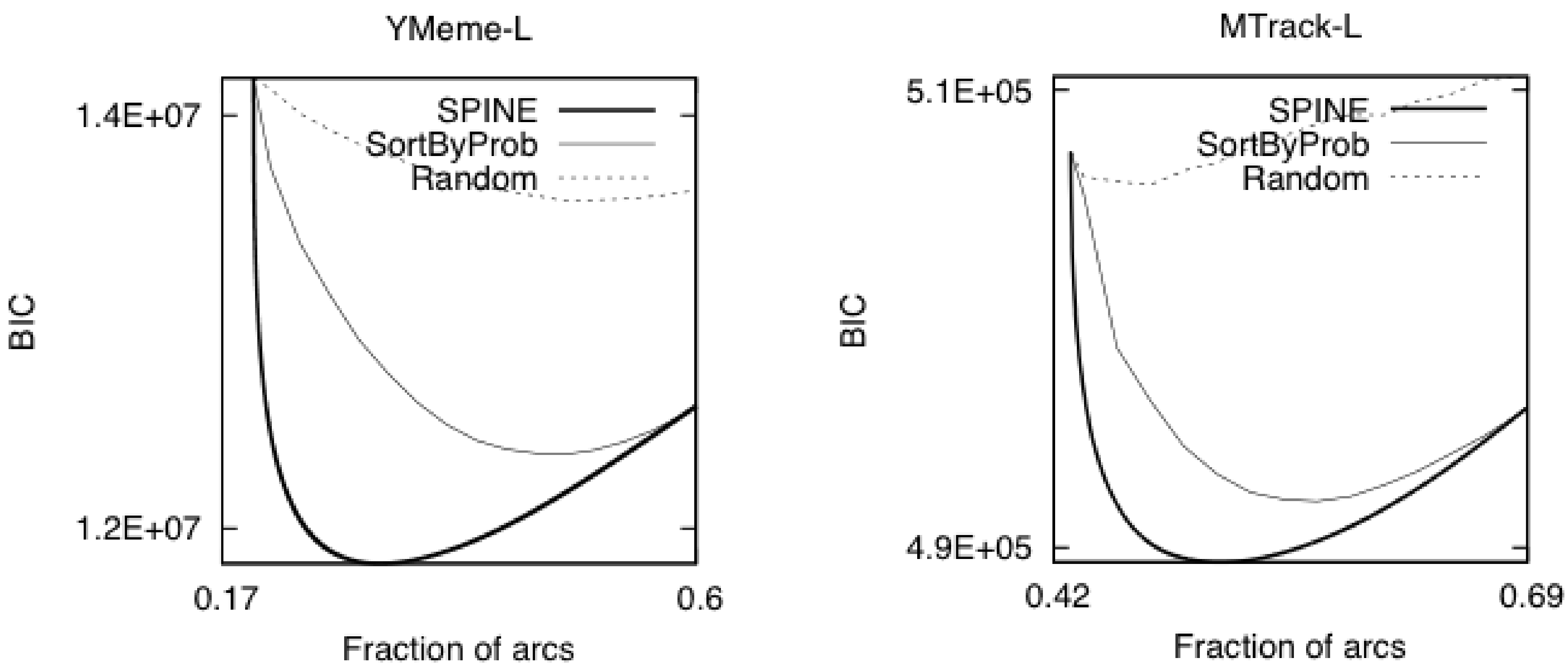
random

Experiments



Model Selection using BIC

BIC(k) = -2logL + klogN



Application

spine as a preprocessing step

influence maximization

select k nodes to maximize spread of action

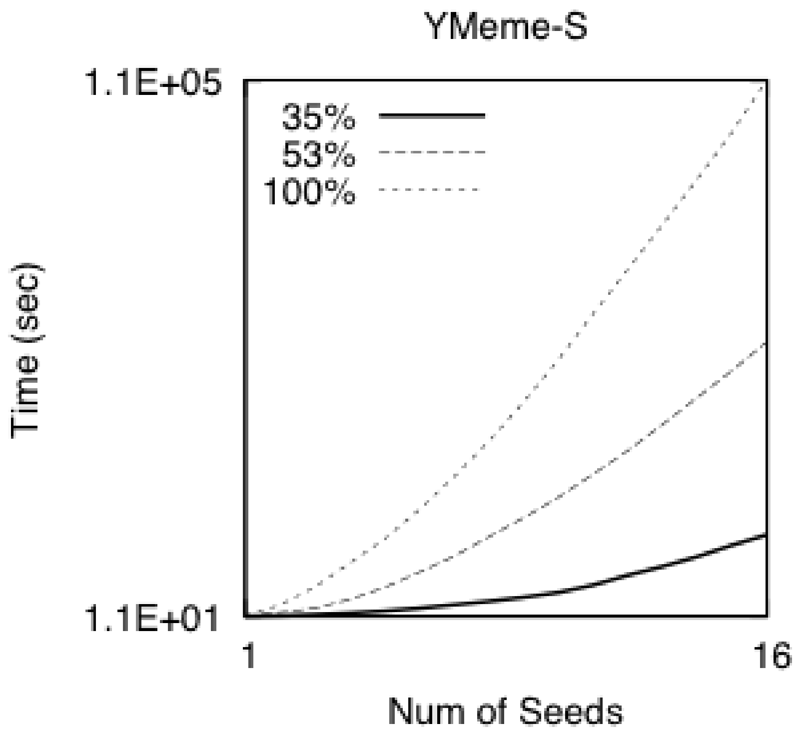
[Kempe, Kleinberg, Tardos, 03]

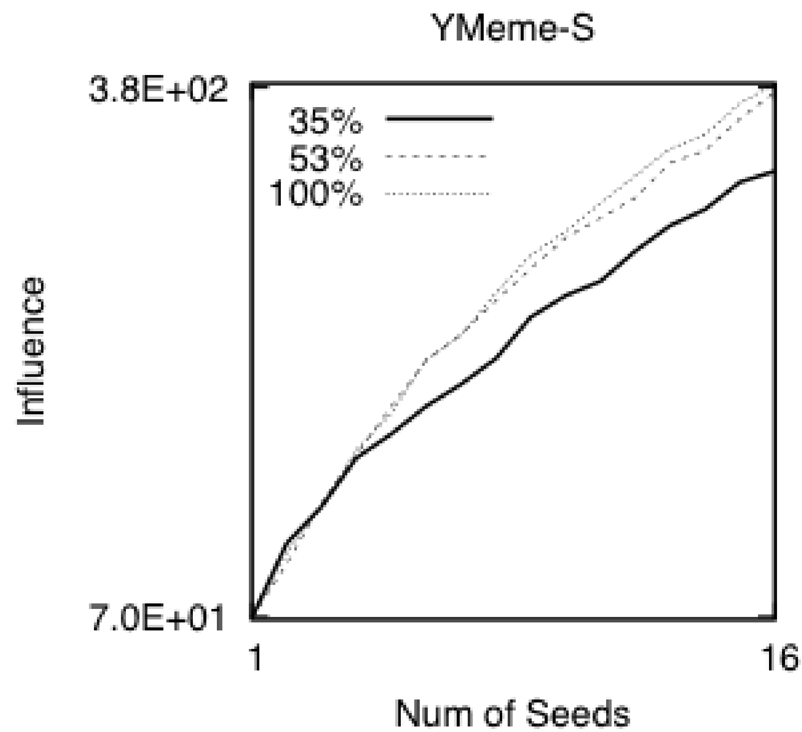
NP-hard, greedy approximation

perform on sparsified network instead

large benefit in efficiency, little loss in quality

Application





Public Code and Data

http://www.cs.toronto.edu/~mathiou/spine/

The End

Questions?