Core Decomposition of Uncertain Graphs

Francesco Bonchi (Yahoo Labs, Barcelona)
Francesco Gullo (Yahoo Labs, Barcelona)
Andreas Kaltenbrunner (Barcelona Media)
Yana Volkovich (Cornell Tech, Barcelona Media)

Introduction

Core Decomposition of Uncertain Graphs

Introduction

Core Decomposition of Uncertain Graphs

Dense subgraphs

- finding dense subgraphs is a fundamental primitive in many graph problems

Dense subgraphs

- finding dense subgraphs is a fundamental primitive in many graph problems
- different definitions of dense subgraphs: cliques, n-cliques, n-clans, k-plexes, k-cores, etc.

Dense subgraphs

- finding dense subgraphs is a fundamental primitive in many graph problems
- different definitions of dense subgraphs: cliques, n-cliques, n-clans, k-plexes, k-cores, etc.
- most of them are computationally prohibitive: **NP-hard** or at least quadratic

k-core decomposition

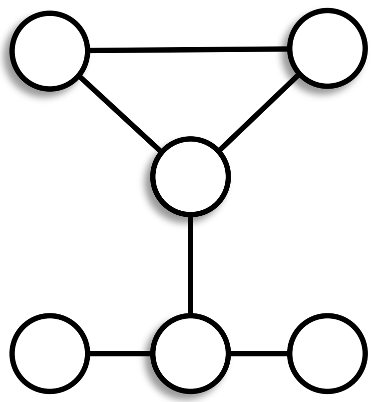
- core decomposition is particularly appealing:
  - it can be computed in linear time
  - it relates to many definitions of dense subgraphs

k-core decomposition

- G =(V, E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$

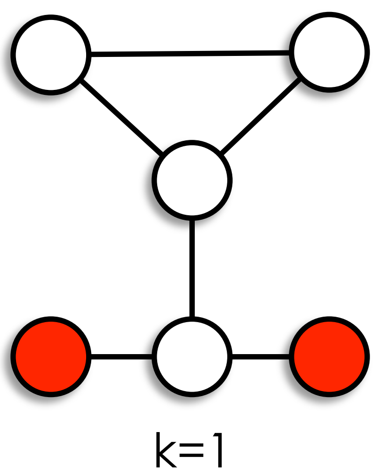
k-core decomposition

- G =(V, E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$



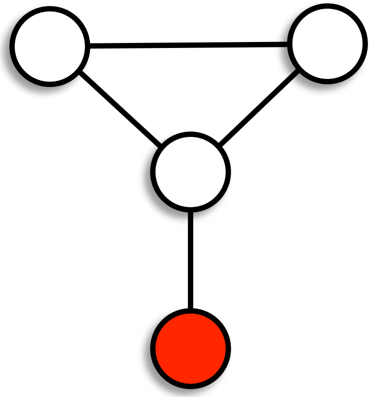
k-core decomposition

- G =(V, E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$



k-core decomposition

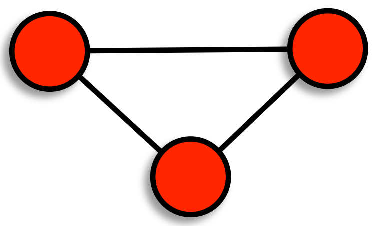
- G =(V, E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$



k=1

k-core decomposition

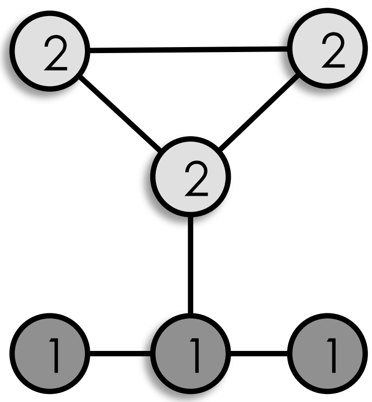
- G =(V, E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$



k=2

k-core decomposition

- G =(V,E) is an undirected graph
- **k-core** of G is a maximal subgraph H = (C, E|C) such that $\forall v \in C : deg_H(v) \geq k$



- **core index** of a vertex v is the highest order of a core that contains v

Introduction

Core Decomposition of Uncertain Graphs

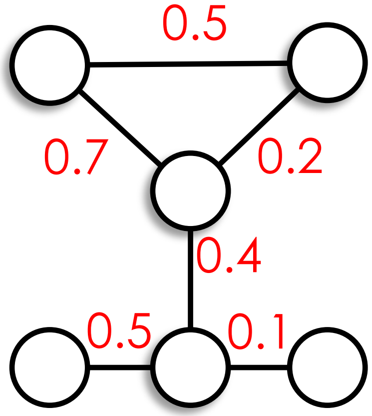
Uncertain graphs

- Many real live networks are associated with uncertainty:
  - data collection process
  - employed machine-learning methods
  - privacy-preserving reasons

- biological networks, protein-interaction networks
- social networks

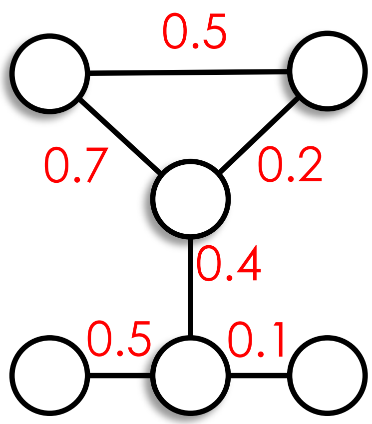
Uncertain graphs

- Edges in an **uncertain graph** are associated with a probability of existence



Uncertain graphs

- Edges in an **uncertain graph** are associated with a probability of existence

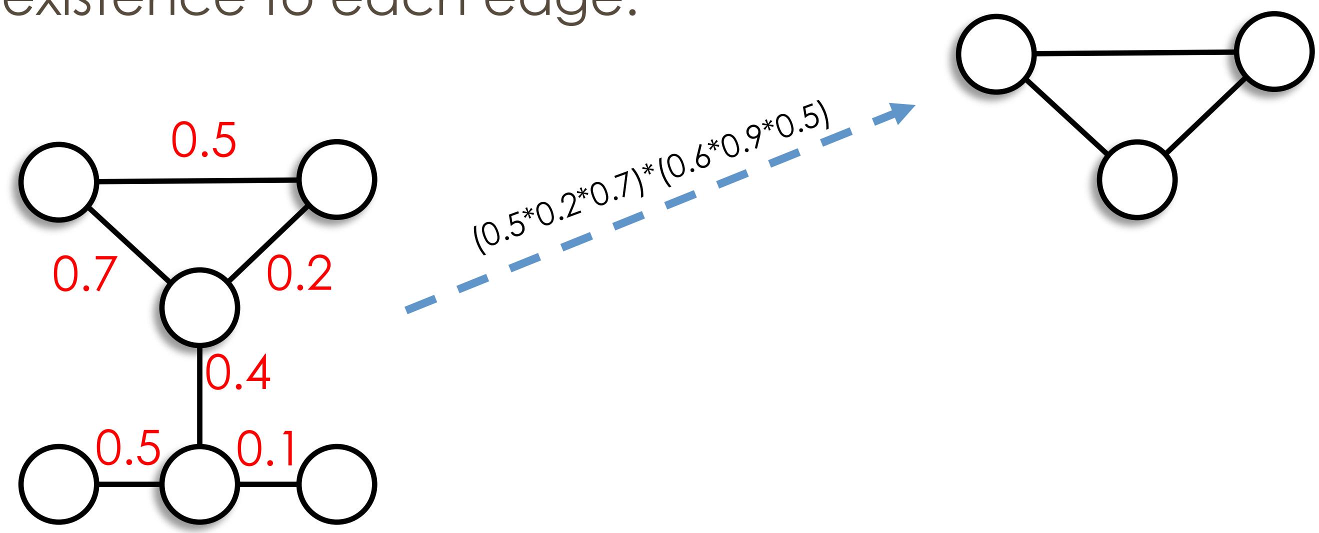


- **Uncertain graph** is a generative model for deterministic graphs

Uncertain graphs

- $\mathcal{G} = (V, E, p)$ be **an uncertain graph**:

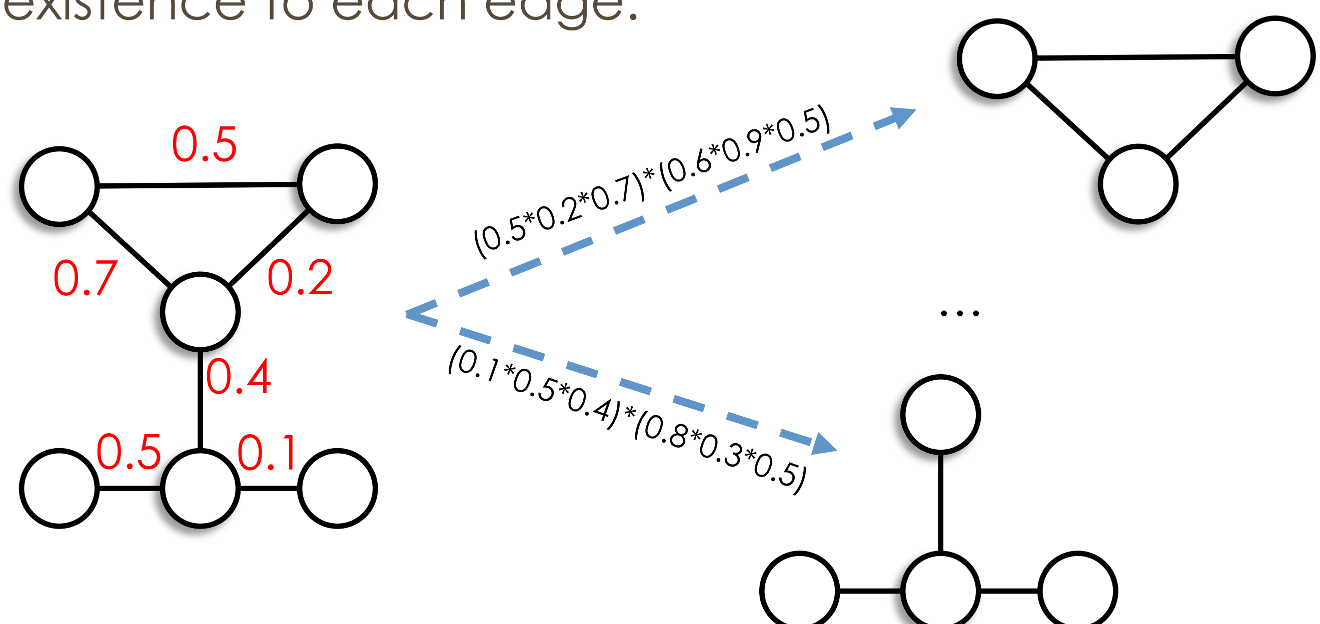
$p : E \rightarrow (0, 1]$ is a function that assigns a probability of existence to each edge.



Uncertain graphs

- $\mathcal{G}$ = (V , E , p) be **an uncertain graph**:

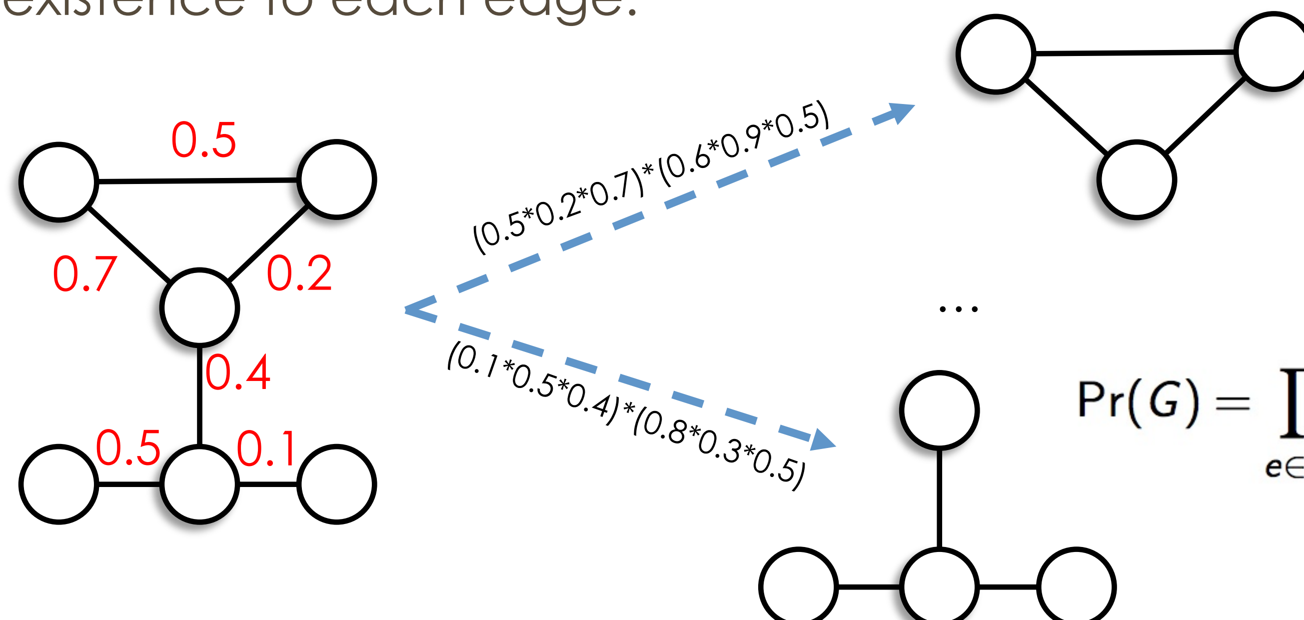
p : E → (0, 1] is a function that assigns a probability of existence to each edge.



Uncertain graphs

- $\mathcal{G}$ = (V , E , p) be **an uncertain graph**:

p : E → (0, 1] is a function that assigns a probability of existence to each edge.



$$\Pr(G) = \prod_{e \in E_G} p_e \prod_{e \in E \setminus E_G} (1 - p_e)$$

Introduction

Core Decomposition of Uncertain Graphs

Introduction

Core Decomposition of Uncertain Graphs

We want to extend the graph tool of core decomposition to the context of uncertain graphs.

Complications

- The fact that core decomposition can be performed in linear time in deterministic graphs does not guarantee efficiency in uncertain graphs.

Complications

- The fact that core decomposition can be performed in linear time in deterministic graphs does not guarantee efficiency in uncertain graphs.

- Are any two vertices connected?

Complications

- The fact that core decomposition can be performed in linear time in deterministic graphs does not guarantee efficiency in uncertain graphs.

- Are any two vertices connected?
  - in deterministic graph: a simple scan of the graph

Complications

- The fact that core decomposition can be performed in linear time in deterministic graphs does not guarantee efficiency in uncertain graphs.

- Are any two vertices connected?
  - in deterministic graph: a simple scan of the graph
  - in uncertain graph: computing the probability that two vertices are connected is a #P-complete problem

Probabilistic (k, η)-cores

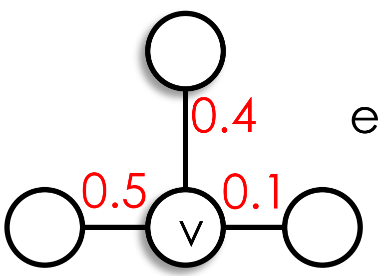
- uncertain graph $\mathcal{G}$ = (V, E, p)
- threshold of uncertainty $\eta \in [0, 1]$

Probabilistic (k, η)-core of $\mathcal{G}$ is a maximal subgraph H = (C, E | C, p) such that $\forall v \in C : \Pr[deg_H(v) \geq k] \geq \eta$

Probabilistic (k, η)-cores

- uncertain graph $\mathcal{G}$ = (V, E, p)
- threshold of uncertainty $\eta \in [0, 1]$

Probabilistic (k, η)-core of $\mathcal{G}$ is a maximal subgraph H = (C, E | C, p) such that $\forall v \in C : Pr[deg_H(v) \geq k] \geq \eta$



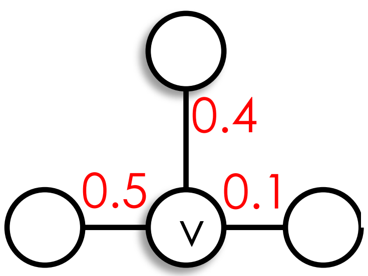
d_v=3

e.g. Pr[deg(v)≥2] =Pr[deg(v)=2] + [deg(v)=3] =
=(0.1*0.5*0.6+0.1*0.4*0.5+0.5*0.4*0.9)+(0.5*0.1*0.4)

Probabilistic (k, η)-cores

- uncertain graph $\mathcal{G}$ = (V, E, p)
- threshold of uncertainty $\eta \in [0, 1]$

Probabilistic (k, η)-core of $\mathcal{G}$ is a maximal subgraph H = (C, E | C, p) such that $\forall v \in C : Pr[deg_H(v) \geq k] \geq \eta$



d_v=3

$$Pr[deg(v) \geq k] = \sum_{i=k}^{d_v} Pr[deg(v) = i] = 1 - \sum_{i=0}^{k-1} Pr[deg(v) = i]$$

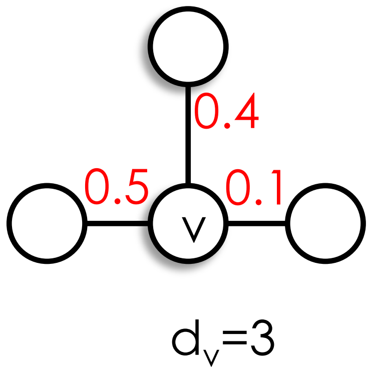
This probability is monotonically non-increasing with k

Probabilistic (k, η)-cores

- **η -degree** of any vertex v ∈ V is defined as
  η -deg(v) = max { k ∈ [0..d_v] | Pr[deg(v) ≥ k] ≥ η }

Probabilistic (k, η)-cores

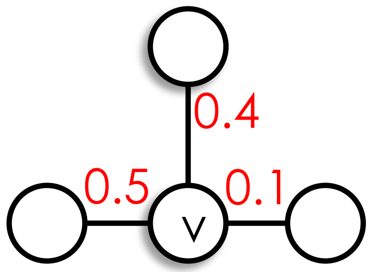
- ***η* -degree** of any vertex $v \in V$ is defined as
  $\eta\text{-deg}(v) = \max \{ k \in [0..d_v] \mid \Pr[\deg(v) \geq k] \geq \eta \}$



η =0.02	η -deg = 3
η =0.25	η -deg = 2
η =0.73	η -deg = 1
η = 1	η -deg = 0

Probabilistic (k, η)-cores

- **η-degree** of any vertex v ∈ V is defined as η-deg(v) = max { k∈[0..d_v] | Pr[deg(v) ≥ k] ≥ η }



d_v=3

η =0.02	η-deg = 3
η =0.25	η-deg = 2
η =0.73	η-deg = 1
η = 1	η-deg = 0

- We use η-degree to define **(k, η)-core decomposition** in a similar manner as degree in deterministic case.

Computing probabilistic cores

- We have proven uniqueness and existence of (k, η)-core decomposition of $\mathcal{G}$.

Computing probabilistic cores

- Since naïve computation of η-degrees leads to exponential time complexity, we defined a dynamic-programming method for (k, η)-core decomposition.

Computing probabilistic cores

- We have shown the running time of (k, η)-core decomposition is **O(mΔ)**, where
  - m is the number of edges
  - Δ is the maximum η-degree over all vertices

Computing probabilistic cores

- We have derived a fast-to-compute lower bound on the η-degree to speed up (k, η)-core computations.

Applications

1. Task-driven team formation
2. Influence-maximization problem

1. Task-driven team formation problem

- A collaboration graph:
  - vertices are individuals
  - edges exhibit a probabilistic topic model representing the topic(s) of past collaborations

1. Task-driven team formation problem

- A collaboration graph:
  - vertices are individuals
  - edges exhibit a probabilistic topic model representing the topic(s) of past collaborations
- A query is a pair ⟨T,Q⟩:
  - T is a set of terms describing a new task
  - Q is a set of vertices

1. Task-driven team formation problem

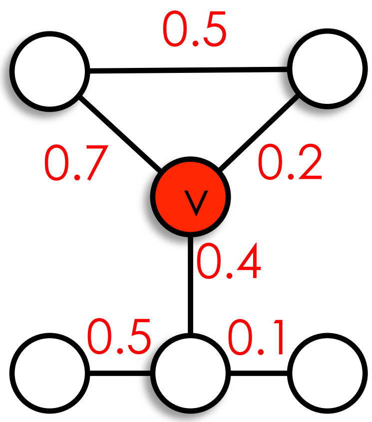
- A collaboration graph:
  - vertices are individuals
  - edges exhibit a probabilistic topic model representing the topic(s) of past collaborations
- A query is a pair $\langle T,Q \rangle$:
  - T is a set of terms describing a new task
  - Q is a set of vertices
- The goal is to find an answer set of vertices A, such that $A \supseteq Q$ is a good team for the task described by T.

2. Influence-maximization problem

- **Independent cascade** (IC) model:
  - Links have associated probability;
  - Every active node v has a single chance of activating each currently inactive neighbor w with probability p_{vw}

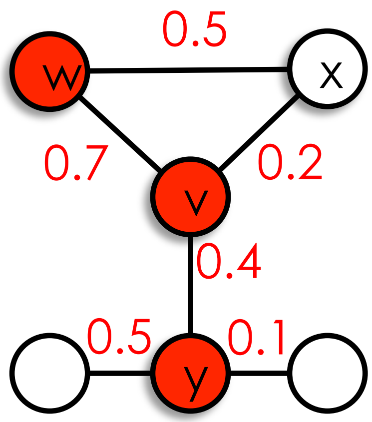
2. Influence-maximization problem

- **Independent cascade** (IC) model:
  - Links have associated probability;
  - Every active node v has a single chance of activating each currently inactive neighbor w with probability p_{vw}



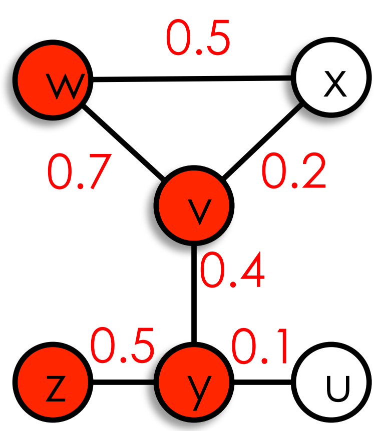
2. Influence-maximization problem

- **Independent cascade** (IC) model:
  - Links have associated probability;
  - Every active node v has a single chance of activating each currently inactive neighbor w with probability p_{vw}



2. Influence-maximization problem

- **Independent cascade** (IC) model:
  - Links have associated probability;
  - Every active node v has a single chance of activating each currently inactive neighbor w with probability p_{vw}



2. Influence-maximization problem

- Finding a set S, $|S|=s$, of vertices that maximizes the expected spread $\sigma(S)$ is a **NP-hard** problem

2. Influence-maximization problem

- Finding a set S, $|S|=s$, of vertices that maximizes the expected spread $\sigma(S)$ is a **NP-hard** problem
- GREEDY algorithm adds the vertex bringing the largest marginal gain in the objective function.

2. Influence-maximization problem

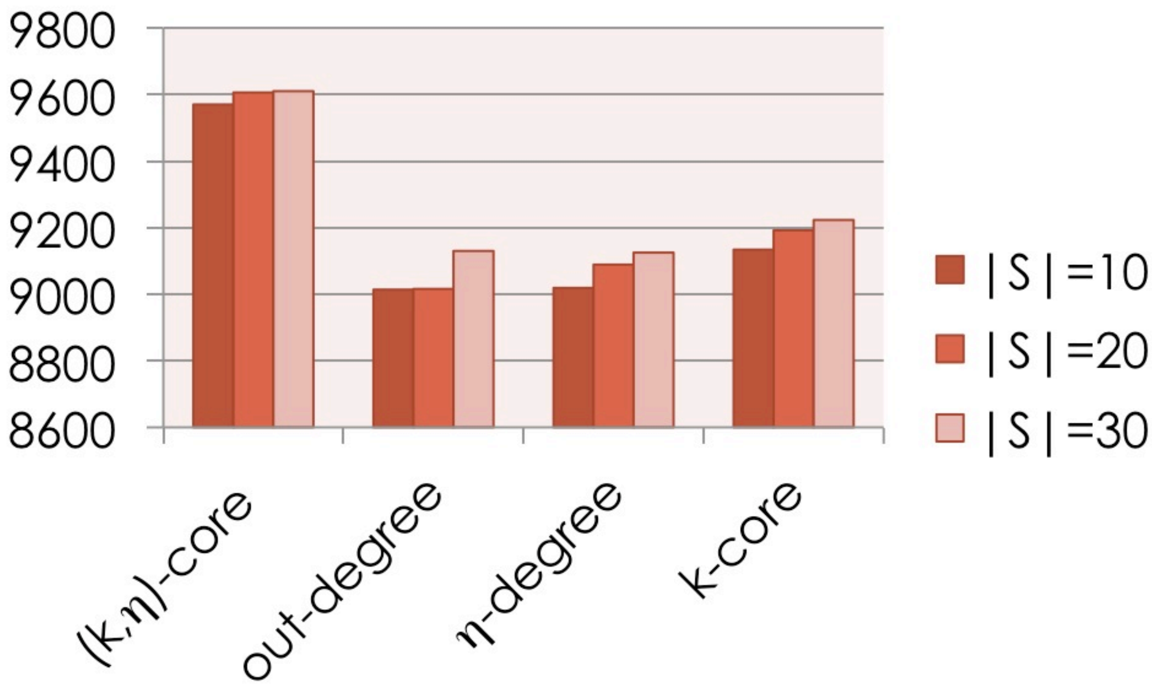
- Finding a set S, $|S|=s$, of vertices that maximizes the expected spread $\sigma(S)$ is a **NP-hard** problem
- GREEDY algorithm adds the vertex bringing the largest marginal gain in the objective function.
- We reduce the input graph G by some rule and run the GREEDY algorithm.

2. Influence-maximization problem

- Finding a set S, $|S|$=s, of vertices that maximizes the expected spread $\sigma(S)$ is a **NP-hard** problem
- GREEDY algorithm adds the vertex bringing the largest marginal gain in the objective function.
- We reduce the input graph $\mathcal{G}$ by some rule and run the GREEDY algorithm.
- On deterministic graph k-core index is a direct indicator of the expected spread of any vertex (experimentally observed).

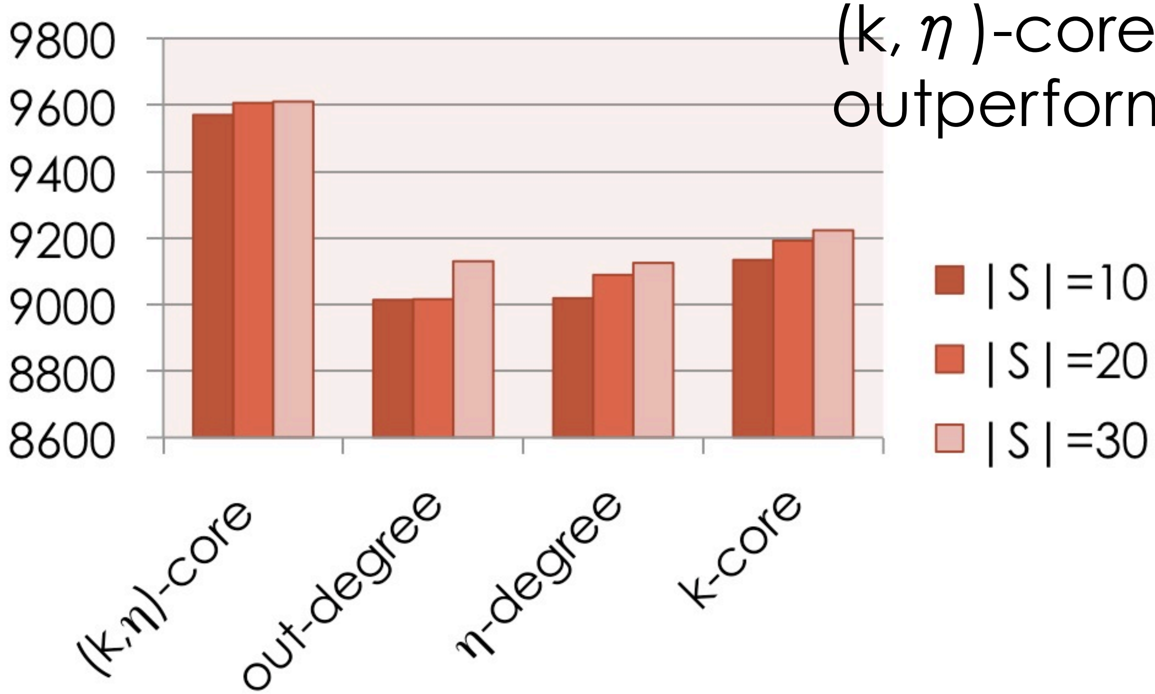
Influence-maximization experiment

- Small directed graph from Twitter with influence probabilities learned from past propagations of URLs (|V| = 21882, |E| = 372005).



Influence-maximization experiment

- Small directed graph from Twitter with influence probabilities learned from past propagations of URLs (|V| = 21882, |E| = 372005).



(k, η)-cores-based method outperforms all the baselines

Conclusions

- We have extended the graph tool of core decomposition to the context of uncertain graphs.

Conclusions

- We have defined the (k, η)-core concept, and devised efficient algorithms for computing a (k, η)-core decomposition.

Conclusions

- We have extensively evaluated our definitions and methods on a number of real-world datasets and applications.

Conclusions

- We plan to investigate the relationship between (k, η)-cores and other definitions of (probabilistic) dense subgraphs.

Questions?