Core Decomposition of Uncertain Graphs

Francesco Bonchi[1], Francesco Gullo[1], Andreas Kaltenbrunner[2] and Yana Volkovich[2]

[1]Yahoo Labs, Barcelona (Spain); [2] Fundació Barcelona Media – Social Media Research Group, Barcelona (Spain)

Abstract

Core decomposition has proven to be a useful primitive for a wide range of graph analyses. One of its most appealing features is that, unlike other notions of dense subgraphs, it can be computed linearly in the size of the input graph.

In this paper we provide an analogous tool for uncertain graphs, i.e., graphs whose edges are assigned a probability of existence. The fact that core decomposition can be computed efficiently in deterministic graphs does not guarantee efficiency in uncertain graphs, where even the simplest graph operations may become computationally intensive. Here we show that core decomposition of uncertain graphs can be carried out efficiently as well.

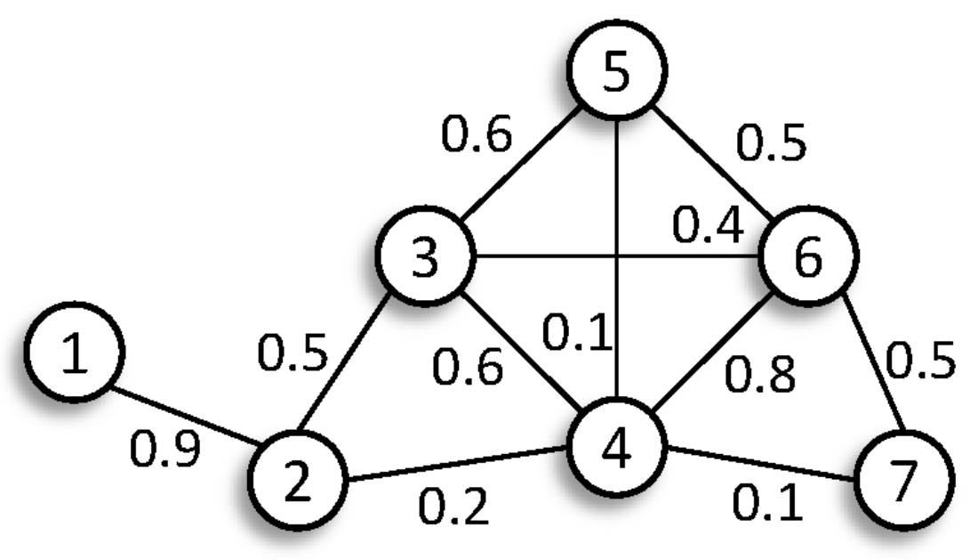
We extensively evaluate our definitions and methods on a number of real-world datasets and applications, such as **influence maximization** and **task-driven team formation**.

Main Findings

- We study the problem of **core decomposition of uncertain graphs**.
- We define the concept of (k,η)-*core* and the corresponding notion of (k,η)-*core* decomposition.
- We devise fast (polynomial-time) algorithms to compute a $((k,\eta)$-*core* decomposition.

Introduction

Uncertain graphs



graphs whose edges are associated with a probability:

- *biological networks; protein-interaction networks*: vertices are genes and/or proteins while edges are interactions.
- *social networks:* edge probabilities may represent the uncertainty (or the accuracy) of link prediction or the influence of one person on another (viral marketing).

Motivation

- Core decomposition can be performed in linear time in deterministic graphs but this does not guarantee efficiency in uncertain graphs.
- E.g., the *two-terminal-reachability* problem (are any two query vertices connected?)
  - in deterministic graphs: a simple scan of the graph
  - in uncertain graphs: computing the probability that two vertices are connected is a $\#\mathbf{P}$-complete problem.

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, denoted $c(v)$, is the highest order of a core that contains v.

Algorithm 1: k-core

```
Input: A graph $G = (V, E)$.
Output: An $n$-dimensional vector $\mathbf{c}$ containing the core number of each $v \in V$.
c ← ∅,   d ← ∅,   D ← [∅, . . . , ∅]
for all $v \in V$ do
    $\mathbf{d}[v] \leftarrow deg(v)$
    $\mathbf{D}[deg(v)] \leftarrow \mathbf{D}[deg(v)] \cup \{v\}$
end for
for all $k = 0, 1, \ldots, n$ do
    while $\mathbf{D}[k] \neq \emptyset$ do
        pick and remove a vertex $v$ from $\mathbf{D}[k]$
        $\mathbf{c}[v] \leftarrow k$
        for all $u : (u, v) \in E$, $\mathbf{d}[u] > k$ do
            move $u$ from $\mathbf{D}[\mathbf{d}[u]]$ to $\mathbf{D}[\mathbf{d}[u] - 1]$
            $\mathbf{d}[u] \leftarrow \mathbf{d}[u] - 1$
        end for
        remove $v$ from $G$
    end while
end for
```

- Iteratively removes the smallest-degree vertex and sets the core number of the removed vertex accordingly.
- Runs in $\mathcal{O}(n + m)$ time.

Problem definition

Definition: uncertain graph

- Let $\mathcal{G} = (V, E, p)$ be an **uncertain graph**, where $p : E \rightarrow (0, 1]$ is a function that assigns a probability of existence to each edge.
- For any vertex $v \in V$, let $N_v = \{(u, v) \in E\}$ denote the set of all edges incident to v, and $d_v = |N_v|$ its size.

Possible-world semantics

- **possible-world semantics:** an uncertain graph $\mathcal{G}$ with m edges as a set of 2^m possible deterministic graphs (worlds), each of which containing a subset of the edges in E.
- an uncertain graph $\mathcal{G} = (V, E, p)$ yields a set of possible graphs $\{G = (V, E_G)\}_{E_G \subseteq E}$, and the probability of observing any possible graph $G = (V, E_G) \sqsubseteq \mathcal{G}$ is:

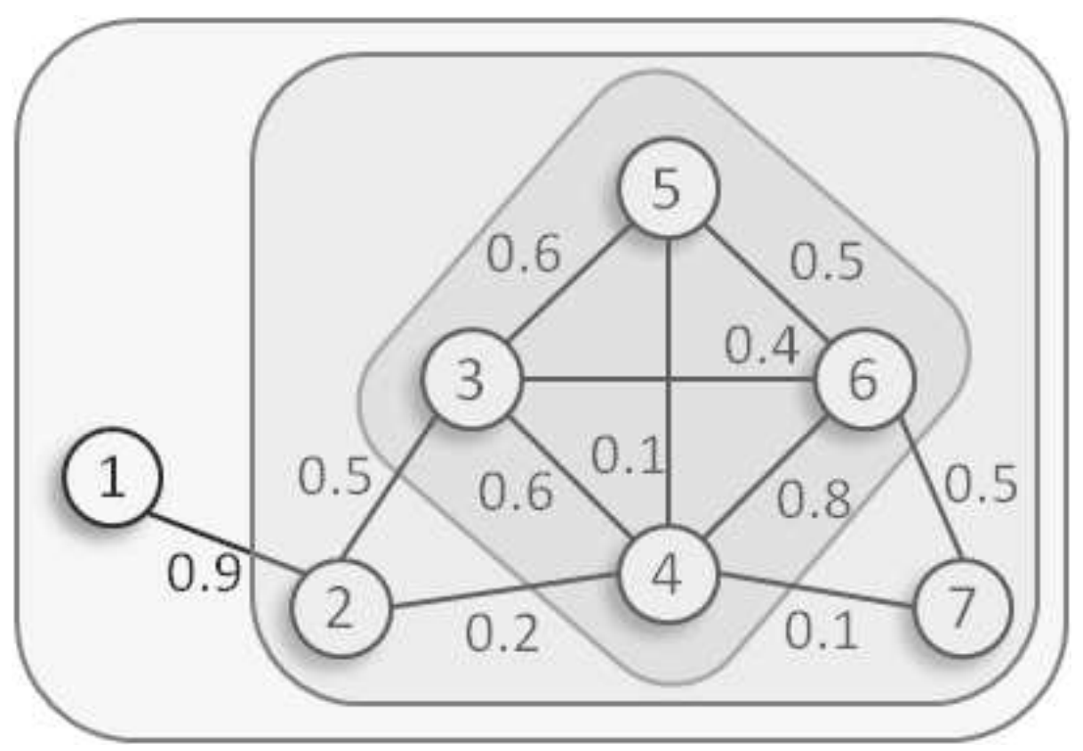
$$\Pr(G) = \prod_{e \in E_G} p_e \prod_{e \in E \setminus E_G} (1 - p_e).$$

Probabilistic (k,η)-cores

- Given an uncertain graph $\mathcal{G} = (V, E, p)$, and a threshold $\eta \in [0, 1]$, the **probabilistic** (k, η)**-core** of $\mathcal{G}$ is a maximal subgraph $\mathcal{H} = (C, E|C, p)$ such that the probability that each vertex $v \in C$ has degree no less than k in $\mathcal{H}$ is greater than or equal to η, i.e., $\forall v \in C : \Pr[deg_{\mathcal{H}}(v) \geq k] \geq \eta$.

Theorem 1 *Given an uncertain graph $\mathcal{G}$ and a probability threshold η, the (k, η)-core decomposition of $\mathcal{G}$ is unique.*

Examples



An uncertain graph and its (k, η)-*core decomposition* for $\eta = 0.04$. Vertex 1 has core number 1, vertices 2 and 7 have core number 2, and vertices 3, 4, 5 and 6 have core number 3.

Computing probabilistic cores

Definition: η-degree

- Given an uncertain graph $\mathcal{G} = (V, E, p)$ and a threshold $\eta \in [0, 1]$, the η**-degree** η-$deg(v)$ of any vertex $v \in V$ is defined as

$$\eta\text{-}deg(v) = \max\{k \in [0..d_v] \mid \Pr[deg(v) \geq k] \geq \eta\}.$$

Let also η-$deg_{\mathcal{H}}(v)$ be the η-degree of v in a subgraph $\mathcal{H}$.

- The η-degree gives an idea of the degree of a vertex given a specific threshold η.
- Idea: exploit η-degree to adapt the k-cores algorithm to uncertain graphs.

Algorithm 2: (k, η)-core

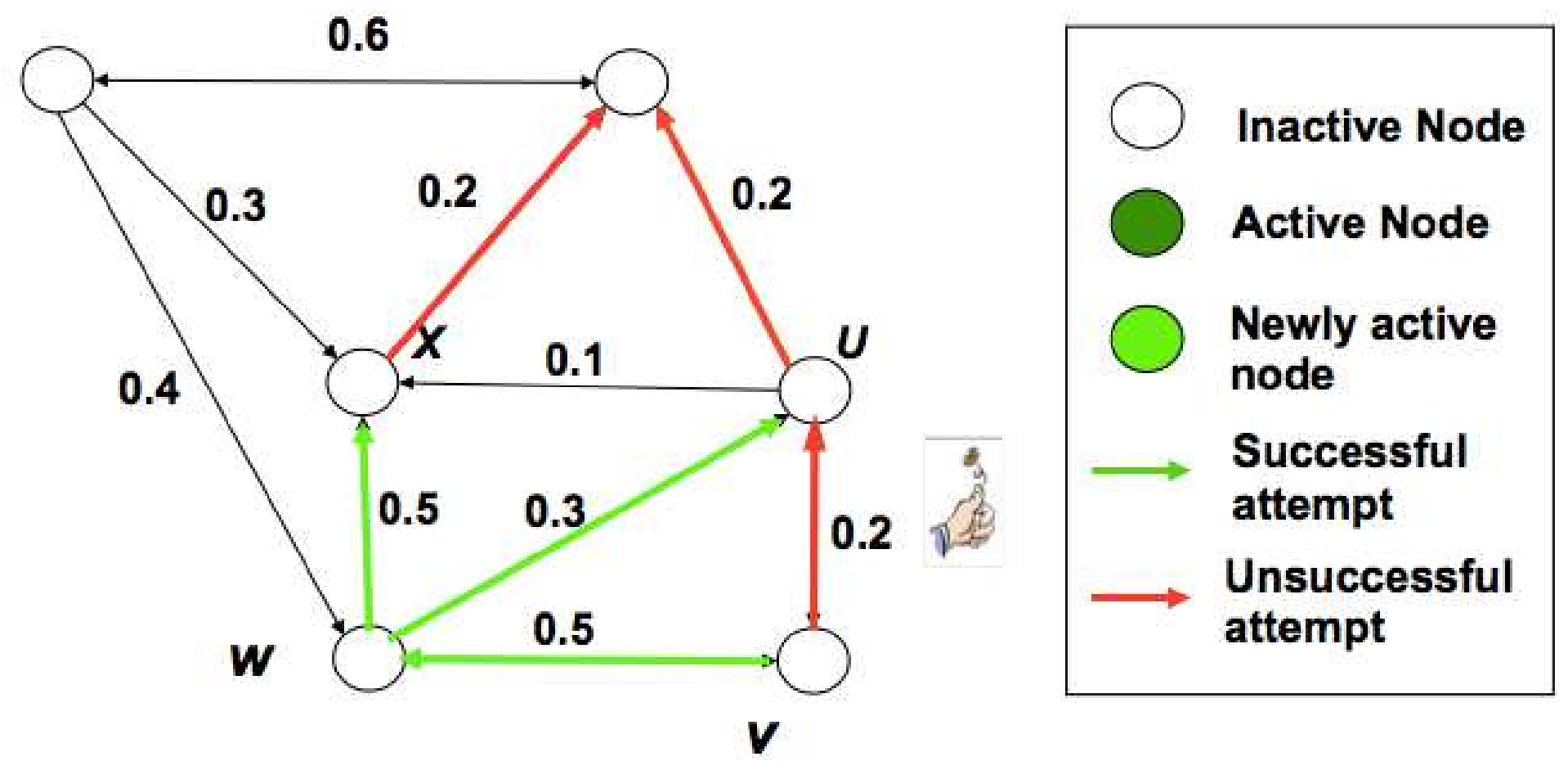
```
Input: An uncertain graph $\mathcal{G} = (V, E, p)$, a threshold $\eta \in [0, 1]$.
Output: An $n$-dimensional vector $\mathbf{c}$ containing the $\eta$-core number of each $v \in V$.
compute $\eta$-$deg(v)$ for all $v \in V$
c ← ∅,   d ← ∅,   D ← [∅, . . . , ∅]
for all $v \in V$ do
    $\mathbf{d}[v] \leftarrow \eta$-$deg(v)$
    $\mathbf{D}[\eta$-$deg(v)] \leftarrow \mathbf{D}[\eta$-$deg(v)] \cup \{v\}$
end for
for all $k = 0, 1, \ldots, n$ do
    while $\mathbf{D}[k] \neq \emptyset$ do
        pick and remove a vertex $v$ from $\mathbf{D}[k]$
        $\mathbf{c}[v] \leftarrow k$
        for all $u : (u, v) \in E$, $\mathbf{d}[u] > k$ do
            recompute $\eta$-$deg(u)$
            move $u$ from $\mathbf{D}[\mathbf{d}[u]]$ to $\mathbf{D}[\eta$-$deg(u)]$
            $\mathbf{d}[u] \leftarrow \eta$-$deg(u)$
        end for
        remove $v$ from $\mathcal{G}$
    end while
end for
```

- The computation of the individual $\Pr[\deg(v) = i]$ values for all $i \in [0..k - 1]$ ($\Pr[deg(v) \geq k]$) can be accomplished in polynomial time (precisely in $\mathcal{O}(kd_v)$ time) adapting [1].
- **Problem:** Numerical instability due to both products and sums of $\tilde{p}_e$ values that can be either very large or very small.
- **Solution:** dynamic-programming method (same time complexity) for efficiently updating η-degrees when an edge is removed.
- **Overall time complexity:** $\mathcal{O}(m\Delta)$ (Δ is the maximum η-degree over all vertices).

Applications

Influence maximization

- Find a set of vertices S, with $|S| = s$, that maximizes the *expected spread*, i.e., the expected number of vertices that would be infected by a viral propagation started in S, under a certain probabilistic propagation model.
- In the *independent cascade model* finding a set S of s vertices that maximizes the expected spread $\sigma(S)$ is **NP**-hard.



Example for the independent cascade model.

- Submodularity of $\sigma(S)$ allows the **Greedy** algorithm that iteratively adds to S the vertex bringing the largest marginal gain in the objective function to achieve $(1 - \frac{1}{e})$ approximation guarantee.
- **Simple idea:** reduce the input graph $\mathcal{G}$ by keeping only the inner-most (k, η)-shells and run the (optimized version of the) Greedy algorithm on such a reduced graph. .
- **Experiment:** A small *directed* graph from Twitter ($|V| = 21\,882$, $|E| = 372\,005$), and a set of propagations of URLs in the social graph, that we use as past evidence to learn the influence probabilities
- Redfuced graph has 2 064 vertices and 86 142 edges.
- Expected spread achieved by the proposed (k,η)-cores-based method vs. some baselines with varying the output set size $|S|$.

	$\|S\| = 10$	$\|S\| = 20$	$\|S\| = 30$
(k,η)-cores	9 570	9 606	9 610
out-degree	9 014	9 016	9 130
η-degree	9 019	9 089	9 125
exp-degree	9 012	9 093	9 123
k-cores	9 134	9 192	9 223

- (k, η)-cores runtime: 4-5 hours;
- baseline runtime: > 1 week.

Task-driven team formation

- Given a collaboration graph $G = (V, E, \tau)$, where vertices are individuals and edges are associated with a probabilistic topic model τ, representing (a distribution on) the topics exhibited by past collaborations.
- Given a collaboration graph $G = (V, E, \tau)$ and a query $\langle T, Q \rangle$, let $\mathcal{G}^T$ be the uncertain graph derived from G and T. Given a threshold $\eta \in [0, 1]$, we want to find a *connected* subgraph $\mathcal{H} = (V_{\mathcal{H}}, E_{\mathcal{H}})$ of $\mathcal{G}^T$ induced by a set of vertices $V_{\mathcal{H}}$ such that

$$V_{\mathcal{H}} = \arg\max_{Q \subseteq S \subseteq V} \min_{u \in S} \eta\text{-}deg(u).$$

- **Algorithm:**
  1. Given a collaboration graph $G = (V, E, \tau)$ and a task-driven team-formation query $\langle T, Q \rangle$, derive the uncertain graph $\mathcal{G}^T = (V, E, p^T)$.
  2. Compute the (k,η)-core decomposition **C** of $\mathcal{G}^T$;
  3. Visit the cores in **C** starting from the smallest-sized one (i.e., the inner-most core), until finding C^*_Q;
  4. Return the connected component of C^*_Q containing Q as the solution.
- **Example:** DBLP bibliographic database: vertices are authors and an edge connects two authors if they have co-authored at least once.
- The resulting graph has $|V| = 1\,089\,442$ and $|E| = 4\,144\,697$.

$T = \{gene, express\}$, $Q = \{H.V.Jagadish\}$	$T = \{xml, tree\}$, $Q = \{H.V.Jagadish, S.Muthukrishnan\}$	$T = \{auction, model\}$, $Q = \{S.Muthukrishnan\}$
Brian D. Athey, Giovanni Scardoni, Kathleen A. Stringer, Venkateshwar G. Keshamouni, Jing Gao, Terry E. Weymouth, Vasudeva Mahavisno, Charles F. Burant, Christopher W. Beecher, Maureen A. Sartor, Alla Karnovsky, Rork Kuick, Zach Wright, James D. Cavalcoli, Gilbert S. Omenn, **H. V. Jagadish**, Carlo Laudanna, Tim Hull, Barbara R. Mirel, V. Glenn Tarcea	**S. Muthukrishnan**, Panagiotis G. Ipeirotis, Lauri Pietarinen **H. V. Jagadish**, Divesh Srivastava, Nick Koudas	Uri Nadav, Noam Nisan, Jon Feldman, Vahab S. Mirrokni, Gagan Aggarwal, Tanmoy Chakraborty, Aranyak Mehta Evdokia Nikolova, **S. Muthukrishnan**, Martin Pal, Clifford Stein, Eyal Even-Dar Florin Constantin, Yishay Mansour

Acknowledgments

- This work was supported by the CENIT program, project CEN-20101037.
- Yana Volkovich acknowledges financial support though the Torres Quevedo Program from the Spanish Ministry of Science and Innovation, co-funded by the European Social Fund.

[1] Xiang H. Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81:457–469, 1994.