Personalized PageRank dimensionality and algorithmic implications

Vijay G. Subramanian

ECE, University of Michigan

Jan 31^{st} 2020, Shannon Channel, Rutgers Univ.

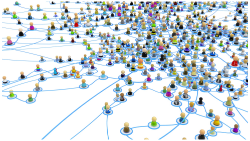
Joint work with Dr. Daniel Vial, ECE, UT Austin & UIUC

Motivation

Graphs arise in many domains, and are used to understand processes, behaviors and vulnerabilities

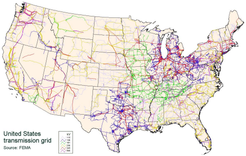
Motivation

Graphs arise in many domains, and are used to understand processes, behaviors and vulnerabilities



Social networks

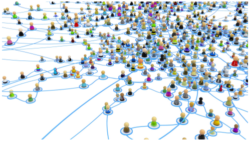
Internet



Power grid

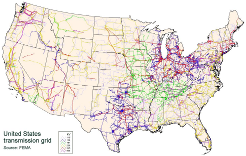
Motivation

Graphs arise in many domains, and are used to understand processes, behaviors and vulnerabilities



Social networks

Internet

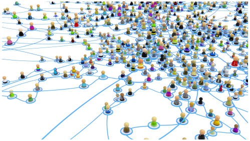


Power grid

Jon Kleinberg (Allerton 2014 plenary): "In the battle of ideas for metaphors for explaining these phenomena, graphs are doing pretty well for themselves."

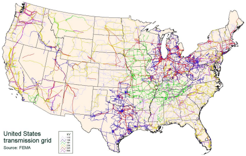
Motivation

Graphs arise in many domains, and are used to understand processes, behaviors and vulnerabilities



Social networks

Internet



Power grid

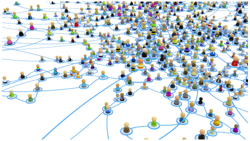
Jon Kleinberg (Allerton 2014 plenary): "In the battle of ideas for metaphors for explaining these phenomena, graphs are doing pretty well for themselves."

Questions of interest in this talk:

- Which nodes are most important/influential, globally and locally?
- Which nodes are similar/relevant to a given node?

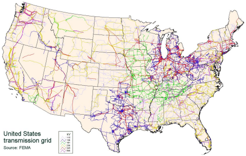
Motivation

Graphs arise in many domains, and are used to understand processes, behaviors and vulnerabilities



Social networks

Internet



Power grid

Jon Kleinberg (Allerton 2014 plenary): "In the battle of ideas for metaphors for explaining these phenomena, graphs are doing pretty well for themselves."

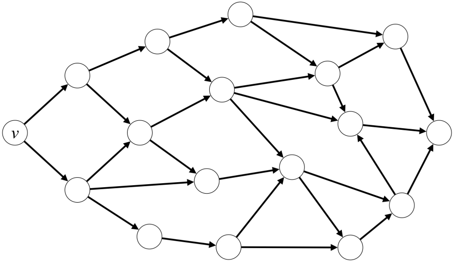
Questions of interest in this talk:

- Which nodes are most important/influential, globally and locally?
- Which nodes are similar/relevant to a given node?

One metric used to answer these questions: Personalized PageRank (PPR)

PPR definition

Given directed graph $G = (V, E)$, let $v \in V$ and $\alpha \in (0, 1)$

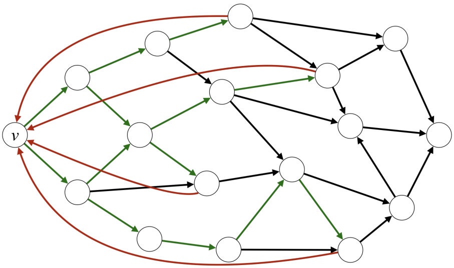


PPR definition

Given directed graph $G = (V, E)$, let $v \in V$ and $\alpha \in (0, 1)$

Define Markov chain $\{X_t^v\}_{t \in \mathbb{N}}$ as follows: given X_t^v,

- W.p. $(1 - \alpha)$, sample X_{t+1}^v from out-neighbors of X_t^v (random walk)
- W.p. α, set $X_{t+1}^v = v$ (jump to v)



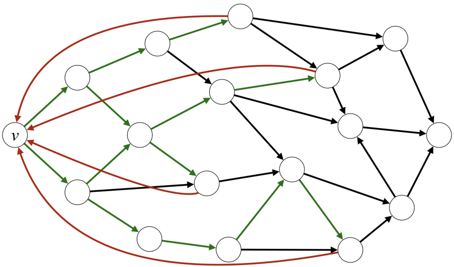
PPR definition

Given directed graph $G = (V, E)$, let $v \in V$ and $\alpha \in (0, 1)$

Define Markov chain $\{X_t^v\}_{t \in \mathbb{N}}$ as follows: given X_t^v,

- W.p. $(1 - \alpha)$, sample X_{t+1}^v from out-neighbors of X_t^v (random walk)
- W.p. α, set $X_{t+1}^v = v$ (jump to v)

Stationary distribution $\pi_v = \{\pi_v(w)\}_{w \in V}$ called *PPR vector*



PPR definition

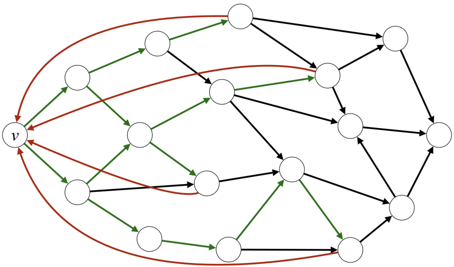
Given directed graph $G = (V, E)$, let $v \in V$ and $\alpha \in (0, 1)$

Define Markov chain $\{X_t^v\}_{t\in\mathbb{N}}$ as follows: given X_t^v,

- W.p. $(1 - \alpha)$, sample X_{t+1}^v from out-neighbors of X_t^v (random walk)
- W.p. α, set $X_{t+1}^v = v$ (jump to v)

Stationary distribution $\pi_v = \{\pi_v(w)\}_{w\in V}$ called *PPR vector*

Matrix $\Pi = \{\pi_v\}_{v\in V}$ called *PPR matrix*



PPR interpretation

$\pi_v(w)$ large when w frequently visited on short walks from v

$\Rightarrow$ Interpret $\pi_v(w)$ as measure of w's importance/relevance to v

PPR interpretation

$\pi_v(w)$ large when w frequently visited on short walks from v

$\Rightarrow$ Interpret $\pi_v(w)$ as measure of w's importance/relevance to v

(Global) PageRank and PPR

PPR interpretation

$\pi_v(w)$ large when w frequently visited on short walks from v

$\Rightarrow$ Interpret $\pi_v(w)$ as measure of w's importance/relevance to v

(Global) PageRank and PPR

Proposed to rank websites (Page et al. 1999); many uses since

- Recommendation (Baluja et al. 2008; Gupta et al. 2013)
- Bioinformatics (Morrison et al. 2005; Freschi 2007)
- Community detection (Andersen, Chung, Lang 2006, Kloumann, Ugander, Kleinberg 2017)
- Graph similarity (Koutra, Vogelstein, Faloutsos 2013)
- ...

Definitions

Directed graph $G = (V, E)$, $V = \{1, 2, \dots, n\}$, $m = |E|$

Adjacency A, diagonal out-degree D, $P = D^{-1}A$ (row stochastic)

PPR: Perron-Frobenius eigenvector π_σ of non-negative matrix

$$P_\sigma = (1 - \alpha)P + \alpha 1_n \sigma^T \qquad (1)$$

where $\alpha \in (0, 1)$, $\sigma \in \mathbb{R}_+^n$ s.t. $\sum_{v \in V} \sigma(v) = 1$ (distribution on V)

When $\sigma = e_s$ (1 in s-th position, 0 elsewhere), denote as π_s

(Global) PageRank: $\sigma = \frac{1}{n} 1_n$

Computation of PPR

- Linear algebraic method
  - Non-negative matrix, so Perron-Frobenius theorem
  - Power method variants, $O(n^2)$ for each source
  - Directed Laplacian variants, almost linear time; Miller, Spielman, Teng, Peng.
- Probabilistic method
  - P is the transition kernel of simple random walk on G
  - Use Monte Carlo (random walks) to estimate PPR
  - $O(n \log(n))$ complexity for (Global) PageRank: Avrachenkov et al. 2007, Sarma et al. 2015
- Variational method
  - View eigenvector computation as a Bellman equation
  - Use value iteration: Andersen, Chung, Lang 2006, Andersen et al. 2008
- Hybrid schemes
  - Monte Carlo + Variational method
  - For a single entry of Π - Lofgren, Banerjee, Goel 2016

Key properties

P_σ is a Doeblin chain: Athreya, Stenflo 2003

$$\pi_s(t) = \mathbb{P}[\text{random walk from } s \text{ of length} \sim \text{geom}(\alpha) \text{ ends at } t] \quad (2)$$

$\Rightarrow$ Can sample from π_s using random walks

To estimate π_σ, suffices to estimate π_s, because

$$\pi_\sigma = \sum_{s \in V} \sigma(s) \pi_s \quad (3)$$

Renewal reward interpretation: $\pi_s(t)$ importance of t for s, as

$$\pi_s(t) \propto \mathbb{E}\left[\text{number of visits to } t \text{ on geom}(\alpha)\text{-length walk from } s\right] \quad (4)$$

PPR dimensionality question

Using Perron-Frobenius theorem, can show $\mathrm{rank}(\Pi) = |V| =: n$

PPR dimensionality question

Using Perron-Frobenius theorem, can show $\text{rank}(\Pi) = |V| =: n$

However, PPR exhibits transitive structure

- $\pi_{v_1}(v_2), \pi_{v_2}(v_3)$ large $\Rightarrow \pi_{v_1}(v_3)$ large ("friend of my friend is my friend")
- Suggests Π has small "effective dimension"

PPR dimensionality question

Using Perron-Frobenius theorem, can show $\text{rank}(\Pi) = |V| =: n$

However, PPR exhibits transitive structure

- $\pi_{v_1}(v_2), \pi_{v_2}(v_3)$ large $\Rightarrow \pi_{v_1}(v_3)$ large ("friend of my friend is my friend")
- Suggests Π has small "effective dimension"

Also, for many real-world graphs $G = (V, E)$, $|E| = O(n)$

- Suggests G is $O(n)$-dimensional, but Π (derived from G) is n^2-dimensional
- Is this gap actually present?

PPR dimensionality question

Using Perron-Frobenius theorem, can show $\text{rank}(\Pi) = |V| =: n$

However, PPR exhibits transitive structure

- $\pi_{v_1}(v_2), \pi_{v_2}(v_3)$ large $\Rightarrow \pi_{v_1}(v_3)$ large ("friend of my friend is my friend")
- Suggests Π has small "effective dimension"

Also, for many real-world graphs $G = (V, E)$, $|E| = O(n)$

- Suggests G is $O(n)$-dimensional, but Π (derived from G) is n^2-dimensional
- Is this gap actually present?

Outline of talk:

1. How to quantify effective dimension of Π?
2. Can we bound this measure of dimensionality?
3. If bound "small", can we leverage it algorithmically?

Quantifying PPR dimensionality

Natural measure of effective dimension of Π:

$$\Delta(\epsilon) = \min_{\hat{\Pi}} \mathrm{rank}(\hat{\Pi}) \text{ s.t. } \|\Pi - \hat{\Pi}\| < \epsilon \tag{5}$$

Intuitively, Π low dimensional if close to low-rank matrix

Quantifying PPR dimensionality

Natural measure of effective dimension of Π:

$$\Delta(\epsilon) = \min_{\hat{\Pi}} \operatorname{rank}(\hat{\Pi}) \text{ s.t. } \|\Pi - \hat{\Pi}\| < \epsilon \tag{5}$$

Intuitively, Π low dimensional if close to low-rank matrix

Can also view (5) as dual of low-rank approximation:

$$\inf_{\hat{\Pi}} \|\Pi - \hat{\Pi}\| \text{ s.t. } \operatorname{rank}(\hat{\Pi}) \leq k$$

Quantifying PPR dimensionality

Natural measure of effective dimension of Π:

$$\Delta(\epsilon) = \min_{\hat{\Pi}} \text{rank}(\hat{\Pi}) \text{ s.t. } \|\Pi - \hat{\Pi}\| < \epsilon \tag{5}$$

Intuitively, Π low dimensional if close to low-rank matrix

Can also view (5) as dual of low-rank approximation:

$$\inf_{\hat{\Pi}} \|\Pi - \hat{\Pi}\| \text{ s.t. } \text{rank}(\hat{\Pi}) \leq k$$

We take $\|\cdot\| = \|\cdot\|_\infty$ in (5), where for matrix A with rows $a_1, \ldots, a_n$,

$$\|A\|_\infty = \max_{i \in \{1,\ldots,n\}} \|a_i\|_1$$

(Natural choice, since $\|\cdot\|_{TV} = \|\cdot\|_1/2$ and each row of Π is a distribution)

Modified dimensionality measure

For analytical/algorithmic reasons, we let $K \subset V$ and upper bound $\Delta(\epsilon)$ as

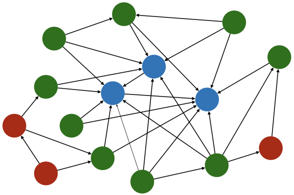
$$\begin{aligned} \Delta(K, \epsilon) &= |K| + \left| \left\{ v \notin K : \min_{\mu_v(k)} \left\| \pi_v - \sum_{k \in K} \mu_v(k) \pi_k \right\|_1 > \epsilon \right\} \right| \\ \mathcal{D}(k, \epsilon) &= K \cup \left\{ v \notin K : \min_{\mu_v(k)} \left\| \pi_v - \sum_{k \in K} \mu_v(k) \pi_k \right\|_1 > \epsilon \right\} \end{aligned} \tag{6}$$

Modified dimensionality measure

For analytical/algorithmic reasons, we let $K \subset V$ and upper bound $\Delta(\epsilon)$ as

$$\begin{aligned} \Delta(K,\epsilon) &= |K| + \left| \left\{ v \notin K : \min_{\mu_v(k)} \left\| \pi_v - \sum_{k \in K} \mu_v(k) \pi_k \right\|_1 > \epsilon \right\} \right| \\ \mathcal{D}(k,\epsilon) &= K \cup \left\{ v \notin K : \min_{\mu_v(k)} \left\| \pi_v - \sum_{k \in K} \mu_v(k) \pi_k \right\|_1 > \epsilon \right\} \end{aligned} \tag{6}$$

- Think of K as hub nodes (located "centrally" in graph)
- Will argue that for most non-hubs, PPR close to linear combo of hub PPR
- Second term in (6) accounts for other non-hubs (typically "far" from hubs)



Brief discussion

Modified dimensionality measure

- Collecting π_v for $v \in \mathcal{D}(K, \epsilon)$ and weights $\boldsymbol{\mu}_v$ for $v \in V$ gives a factorization of Π as H, W
  What are the dimensions? Can this be generated fast?

Brief discussion

Modified dimensionality measure

- Collecting π_v for $v \in \mathcal{D}(K, \epsilon)$ and weights $\boldsymbol{\mu}_v$ for $v \in V$ gives a factorization of Π as H, W
  What are the dimensions? Can this be generated fast?

Would like K to be easily identified too
Can we take nodes that will be visited “first” by random walks?

Graph model

$\Delta(K, \epsilon)$ highly dependent on local graph structure – hard to bound in general

[1] "Nice" $=$ well-approximated by a certain branching process, e.g. Chen, Olvera-Cravioto 2013; Chen, Litvak, Olvera-Cravioto 2017

Graph model

$\Delta(K, \epsilon)$ highly dependent on local graph structure – hard to bound in general

We analyze directed configuration model (DCM) due to “nice” local structure[1]

[1] “Nice” = well-approximated by a certain branching process, e.g. Chen, Olvera-Cravioto 2013; Chen, Litvak, Olvera-Cravioto 2017

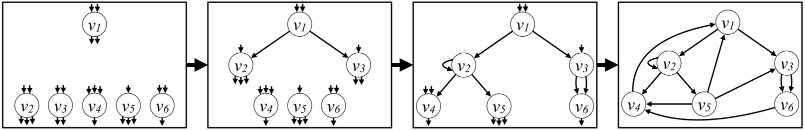
Graph model

$\Delta(K, \epsilon)$ highly dependent on local graph structure – hard to bound in general

We analyze directed configuration model (DCM) due to "nice" local structure[1]

DCM construction:

1. Realize degree sequence $\{d_{out}(v), d_{in}(v)\}_{v \in V}$
2. Attach $d_{out}(v)$ ($d_{in}(v)$, resp.) outgoing (incoming, resp.) half-edges to v
3. Randomly pair half-edges to form edges via breadth-first-search

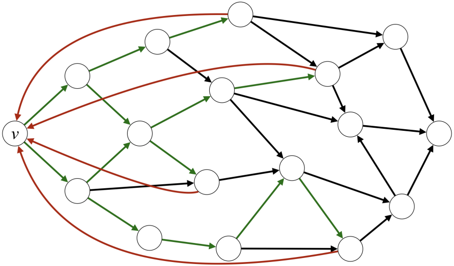


[1] "Nice" = well-approximated by a certain branching process, e.g. Chen, Olvera-Cravioto 2013; Chen, Litvak, Olvera-Cravioto 2017

Jump probability and dimensionality

Choice of $\alpha = \mathbb{P}(\text{jump to } v)$ impacts dimensionality:

- $\alpha \approx 0 \Rightarrow \pi_v \approx$ random walk stationary distribution $\Rightarrow \Delta(K, \epsilon) \approx 1$
- $\alpha \approx 1 \Rightarrow \pi_v \approx$ point mass on $v \Rightarrow \Delta(K, \epsilon) \approx n$

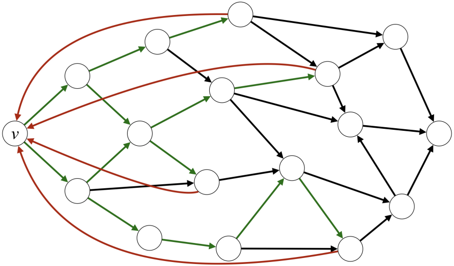


Jump probability and dimensionality

Choice of $\alpha = \mathbb{P}(\text{jump to } v)$ impacts dimensionality:

- $\alpha \approx 0 \Rightarrow \pi_v \approx$ random walk stationary distribution $\Rightarrow \Delta(K, \epsilon) \approx 1$
- $\alpha \approx 1 \Rightarrow \pi_v \approx$ point mass on $v \Rightarrow \Delta(K, \epsilon) \approx n$

How to make this precise?



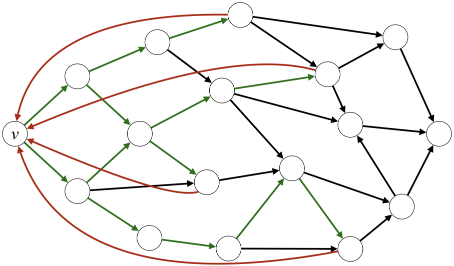
Jump probability and dimensionality

Choice of $\alpha = \mathbb{P}(\text{jump to } v)$ impacts dimensionality:

- $\alpha \approx 0 \Rightarrow \pi_v \approx$ random walk stationary distribution $\Rightarrow \Delta(K, \epsilon) \approx 1$
- $\alpha \approx 1 \Rightarrow \pi_v \approx$ point mass on $v \Rightarrow \Delta(K, \epsilon) \approx n$

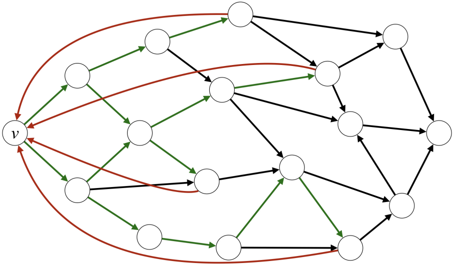
How to make this precise?

Namely, for a sequence $\{G_n\}_{n \in \mathbb{N}}$ of DCMs, how should $\alpha = \alpha_n$ scale with n?



Jump probability and mixing times

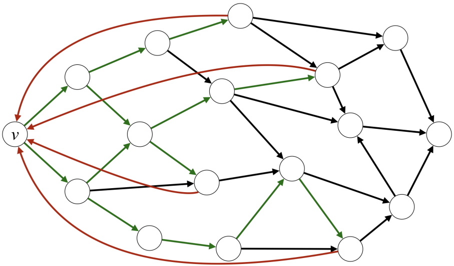
Suppose $\alpha_n \log n \to 0$, e.g. $\alpha_n = 1/(\log n)^2$



Jump probability and mixing times

Suppose $\alpha_n \log n \to 0$, e.g. $\alpha_n = 1/(\log n)^2$

Since $\alpha_n = \mathbb{P}(\text{jump to } v)$, $\mathbb{E}[\text{random walk length}] = \Theta((\log n)^2)$

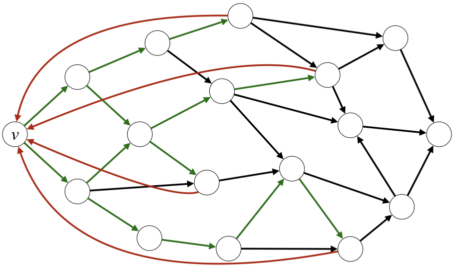


Jump probability and mixing times

Suppose $\alpha_n \log n \to 0$, e.g. $\alpha_n = 1/(\log n)^2$

Since $\alpha_n = \mathbb{P}(\text{jump to } v)$, $\mathbb{E}[\text{random walk length}] = \Theta((\log n)^2)$

Bordenave, Caputo, Salez 2018: random walk on DCM mixes in $\Theta(\log n)$ steps



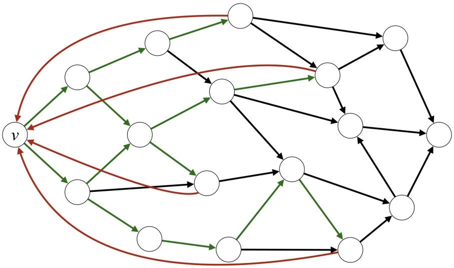
Jump probability and mixing times

Suppose $\alpha_n \log n \to 0$, e.g. $\alpha_n = 1/(\log n)^2$

Since $\alpha_n = \mathbb{P}(\text{jump to } v)$, $\mathbb{E}[\text{random walk length}] = \Theta((\log n)^2)$

Bordenave, Caputo, Salez 2018: random walk on DCM mixes in $\Theta(\log n)$ steps

Mixing occurs before jump to v! Allows us to show $\Delta(K, \epsilon) = 1$ with high prob.



Jump probability and mixing times

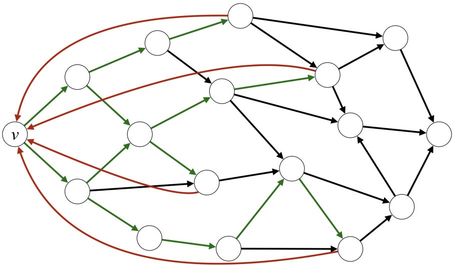
Suppose $\alpha_n \log n \to 0$, e.g. $\alpha_n = 1/(\log n)^2$

Since $\alpha_n = \mathbb{P}(\text{jump to } v)$, $\mathbb{E}[\text{random walk length}] = \Theta((\log n)^2)$

Bordenave, Caputo, Salez 2018: random walk on DCM mixes in $\Theta(\log n)$ steps

Mixing occurs before jump to v! Allows us to show $\Delta(K, \epsilon) = 1$ with high prob.

Hence, we set $\alpha_n = \Theta(1/\log n)$ (just outside the trivial regime)



Brief discussion

Random walk and PPR properties

1. If $\alpha_n =$ constant, then fixed PPR set around any node is constant-sized.
2. If $\alpha_n = \Theta(1/\log n)$, then fixed PPR set around any node increases as n^γ.
3. Scaling also related to the Cheeger number/isoperimetric number of graph family.
4. Recent results of Caputo and Quattroppani also suggest that dimension will be degenerate for any other scaling.
5. Bordenave, Caputo, Salez 2018: Random walk stationary distribution unknown but close in a strong-sense to normalized in-degree distribution.
6. Using high in-degree nodes as hubs will be good.
7. Other choices: high (Global) PageRank but needs a computation.

Main result

Main result concerns sequence of DCMs $\{G_n\}_{n \in \mathbb{N}}$, where G_n has n nodes

Main result

Main result concerns sequence of DCMs $\{G_n\}_{n\in\mathbb{N}}$, where G_n has n nodes

From G_n, define $\Delta_n(K_n, \epsilon)$ for specific K_n (random variable, as G_n is random)

Main result

Main result concerns sequence of DCMs $\{G_n\}_{n\in\mathbb{N}}$, where G_n has n nodes

From G_n, define $\Delta_n(K_n,\epsilon)$ for specific K_n (random variable, as G_n is random)

Our main result says $\Delta_n(K_n,\epsilon) = o(n)$ with high probability as $n\to\infty$:

Theorem

Assume degree sequence satisfies certain assumptions (details to come), and assume $\alpha_n = \Theta(1/\log n)$. Then for any $\epsilon > 0$, some $c_\epsilon \in (0,1)$, and any $C > 0$, all independent of n,

$$\lim_{n\to\infty} \mathbb{P}\left(\Delta_n(K_n,\epsilon) > Cn^{c_\epsilon}\right) = 0.$$

Proof of main result

Main result follows almost immediately from key lemma:

Lemma

Under assumptions of theorem, we have for $s \sim V$ uniformly and some $\tilde{c}_\epsilon > 0$,

$$\mathbb{P}\left(\underbrace{\min_{\mu_s(k)} \left\| \pi_s - \sum_{k \in K} \mu_s(k) \pi_k \right\|_1}_{\star} > \epsilon\right) = O\left(n^{-\tilde{c}_\epsilon}\right).$$

Proof of main result

Main result follows almost immediately from key lemma:

Lemma

Under assumptions of theorem, we have for $s \sim V$ uniformly and some $\tilde{c}_\epsilon > 0$,

$$\mathbb{P}\left(\min_{\mu_s(k)} \underbrace{\left\| \pi_s - \sum_{k \in K} \mu_s(k) \pi_k \right\|_1}_{\star} > \epsilon \right) = O\left(n^{-\tilde{c}_\epsilon} \right).$$

Outline for proof of lemma:

1. Show $\star$ depends only on neighborhood of s **for certain** $\mu_s(k)$
2. Approximate neighborhood construction with branching process (using Chen, Litvak, Olvera-Cravioto 2017) to study $\star$ on tree
3. Recursive nature of branching process $\rightarrow$ $\star$ on tree is martingale-like $\rightarrow$ analyze similar to method of bounded differences

Choice of $\mu_v(k)$

By considering first step of PPR Markov chain, can show

$$\pi_v(w) = \underbrace{\alpha 1(w = v)}_{\text{first step is jump to } v} + \underbrace{\sum_{k:v \to k} \frac{(1-\alpha)}{|\{k : v \to k\}|} \pi_k(w)}_{\text{first step follows random walk}}$$

Choice of $\mu_v(k)$

By considering first step of PPR Markov chain, can show

$$\pi_v(w) = \underbrace{\alpha 1(w = v)}_{\text{first step is jump to } v} + \underbrace{\sum_{k:v \to k} \frac{(1-\alpha)}{|\{k : v \to k\}|} \pi_k(w)}_{\text{first step follows random walk}}$$

For any $K \subset V$, Jeh, Widom 2003 proves decomposition of same form:

$$\pi_v(w) = \frac{\alpha 1(w \notin K)\tilde{\pi}_v(w)}{\alpha + (1-\alpha)\tilde{\pi}_v(K)} + \sum_{k \in K} \frac{\tilde{\pi}_v(k)}{\alpha + (1-\alpha)\tilde{\pi}_v(K)} \pi_k(w)$$

where $\tilde{\pi}_v$ is PPR on graph with outgoing edges from K removed

Choice of $\mu_v(k)$

By considering first step of PPR Markov chain, can show

$$\pi_v(w) = \underbrace{\alpha 1(w = v)}_{\text{first step is jump to } v} + \underbrace{\sum_{k:v \to k} \frac{(1-\alpha)}{|\{k : v \to k\}|} \pi_k(w)}_{\text{first step follows random walk}}$$

For any $K \subset V$, Jeh, Widom 2003 proves decomposition of same form:

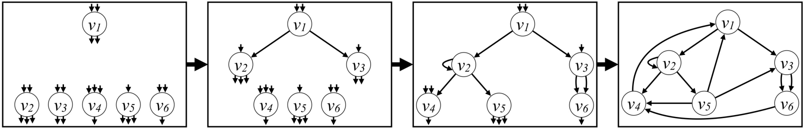
$$\pi_v(w) = \frac{\alpha 1(w \notin K)\tilde{\pi}_v(w)}{\alpha + (1-\alpha)\tilde{\pi}_v(K)} + \sum_{k \in K} \frac{\tilde{\pi}_v(k)}{\alpha + (1-\alpha)\tilde{\pi}_v(K)} \pi_k(w)$$

where $\tilde{\pi}_v$ is PPR on graph with outgoing edges from K removed

In proof (and later, in algorithm), we let $\mu_v(k) = \frac{\tilde{\pi}_v(k)}{\alpha+(1-\alpha)\tilde{\pi}_v(K)}$

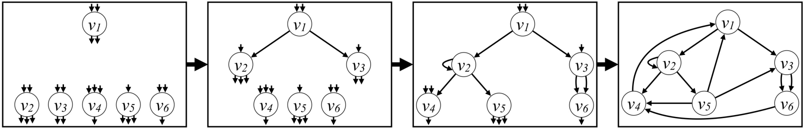
Assumptions (1/2)

Recall: DCM randomly pairs edges from degree sequence $\{d_{out}(v), d_{in}(v)\}_{v \in V}$



Assumptions (1/2)

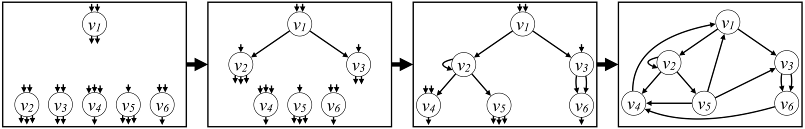
Recall: DCM randomly pairs edges from degree sequence $\{d_{out}(v), d_{in}(v)\}_{v \in V}$



We assume $\{d_{out}(v), d_{in}(v)\}_{v \in V}$ satisfies two properties with high probability

Assumptions (1/2)

Recall: DCM randomly pairs edges from degree sequence $\{d_{out}(v), d_{in}(v)\}_{v \in V}$



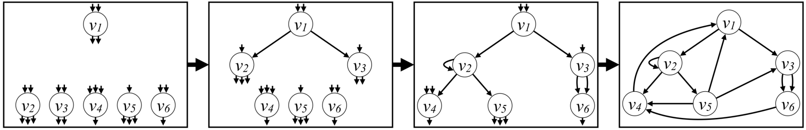
We assume $\{d_{out}(v), d_{in}(v)\}_{v \in V}$ satisfies two properties with high probability

Property 1: $\{d_{out}(v), d_{in}(v)\}_{v \in V}$ is sparse (e.g. $O(n)$ total edges)

$\Rightarrow$ Needed for branching process approximation; possible artifact of analysis

Assumptions (1/2)

Recall: DCM randomly pairs edges from degree sequence $\{d_{out}(v), d_{in}(v)\}_{v \in V}$



We assume $\{d_{out}(v), d_{in}(v)\}_{v \in V}$ satisfies two properties with high probability

Property 1: $\{d_{out}(v), d_{in}(v)\}_{v \in V}$ is sparse (e.g. $O(n)$ total edges)

$\Rightarrow$ Needed for branching process approximation; possible artifact of analysis

Property 2: $|K| = o(n)$ but K contains non-vanishing fraction of edges, i.e.

$$\frac{\sum_{k \in K} d_{in}(k)}{\sum_{v \in V} d_{in}(v)} \xrightarrow[n \to \infty]{} p > 0$$

$\Rightarrow$ We believe this assumption is **fundamentally necessary**

Assumptions (2/2)

Recall key property:

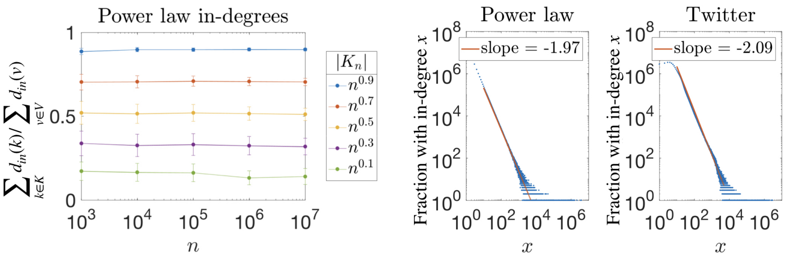
$$|K| = o(n), \quad \frac{\sum_{k \in K} d_{in}(k)}{\sum_{v \in V} d_{in}(v)} \xrightarrow[n \to \infty]{} p > 0 \tag{7}$$

Assumptions (2/2)

Recall key property:

$$|K| = o(n), \quad \frac{\sum_{k \in K} d_{in}(k)}{\sum_{v \in V} d_{in}(v)} \xrightarrow[n \to \infty]{} p > 0 \qquad (7)$$

Empirically holds if $d_{in}(v)$ follow power law, common model for e.g. Twitter



Geometric interpretation of theorem

Theorem says for most $v \notin K$ and some $\mu_v(k) \geq 0$,

$$\pi_v \approx \sum_{k \in K} \mu_v(k) \pi_k$$

Geometric interpretation of theorem

Theorem says for most $v \notin K$ and some $\mu_v(k) \geq 0$,

$$\pi_v \approx \sum_{k \in K} \mu_v(k) \pi_k$$

When $|V|$ large, we also show $\sum_{k \in K} \mu_v(k) \approx 1$, so for most $v \notin K$,

$$\pi_v \approx \text{convex combination of } \{\pi_k\}_{k \in K}$$

Geometric interpretation of theorem

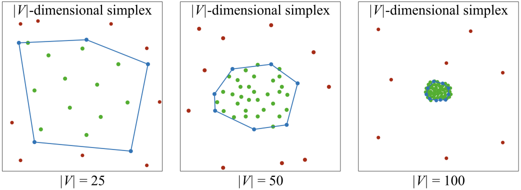
Theorem says for most $v \notin K$ and some $\mu_v(k) \geq 0$,

$$\pi_v \approx \sum_{k \in K} \mu_v(k) \pi_k$$

When $|V|$ large, we also show $\sum_{k \in K} \mu_v(k) \approx 1$, so for most $v \notin K$,

$$\pi_v \approx \text{convex combination of } \{\pi_k\}_{k \in K}$$

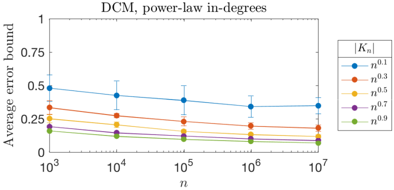
$\Rightarrow$ Most of $\{\pi_v\}_{v \notin K}$ lie near convex hull of $\{\pi_k\}_{k \in K}$, which shrinks relative to $|V|$-dimensional simplex (a few $\{\pi_v\}_{v \notin K}$ can be far away)



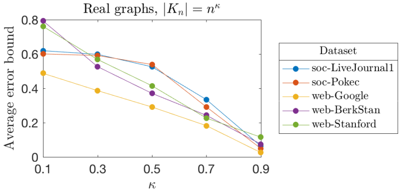
Empirical results (1/2)

Compute bound on $\|\pi_v - \sum_{k \in K} \mu_v(k) \pi_k\|_1$, averaged across $v \notin K$

Set K = nodes of highest in-degree, $\alpha_n = 1/\log n$



For DCM with power law in-degrees, average error decays as n grows (despite $|K|/n$ decaying too)



For variety of real graphs, average error decays as κ grows when $K = n^\kappa$ nodes of highest in-degree

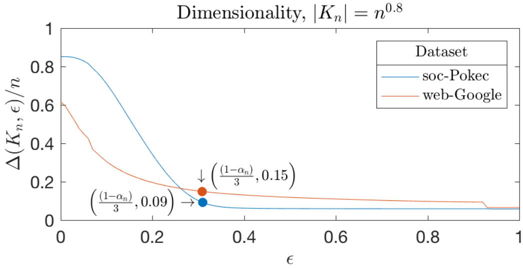
Empirical results (2/2)

Bound $\Delta(K, \epsilon)$ for two real graphs (social network, partial web crawl)

K and α_n chosen as in previous slide

For soc-Pokec, $\Delta(K, \epsilon) = 0.09n$ when $\epsilon = \frac{1-\alpha_n}{3}$; similar for web-Google[2]

Thus, while theorem doesn't apply, $\Delta(K, \epsilon)$ small relative to n for reasonable ϵ

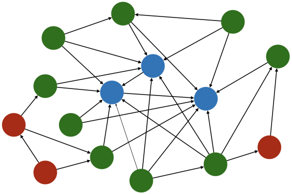


[2] Can show worst-case error is $1 - \alpha_n$, so this ϵ reduces worst-case by factor of 3

Baseline algorithm (Jeh, Widom 2003)

Jeh, Widom 2003 proposes (but doesn't analyze!) the following:

1. Choose "hub" nodes, estimate PPR vectors directly
2. For other nodes, estimate PPR as linear combo of hub PPR[3]



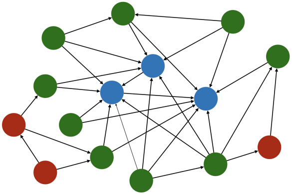
[3] Using decomposition shown previously

Baseline algorithm (Jeh, Widom 2003)

Jeh, Widom 2003 proposes (but doesn't analyze!) the following:

1. Choose "hub" nodes, estimate PPR vectors directly
2. For other nodes, estimate PPR as linear combo of hub PPR[3]

Our result ⇒ linear combo good estimate for all but $o(n)$ non-hubs if $o(n)$ hubs



[3] Using decomposition shown previously

Baseline algorithm (Jeh, Widom 2003)

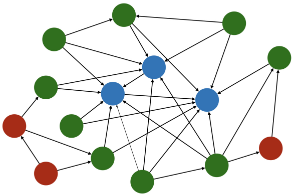
Jeh, Widom 2003 proposes (but doesn't analyze!) the following:

1. Choose "hub" nodes, estimate PPR vectors directly
2. For other nodes, estimate PPR as linear combo of hub PPR[3]

Our result $\Rightarrow$ linear combo good estimate for all but $o(n)$ non-hubs if $o(n)$ hubs

Thus, we improve Jeh, Widom 2003, but questions remain:

- Can we guarantee accuracy *all* nodes?
- Can we estimate hub PPR, and non-hub linear combo weights, with **provably good performance?** (good heuristics such as Global PageRank in Jeh, Widom 2003)



[3] Using decomposition shown previously

Improving accuracy of baseline scheme

Baseline scheme: for $v \notin K$, π_v estimated as

$$\hat{\pi}_v = \sum_{k \in K} \mu_v(k) \pi_k$$

where $\mu_v(k)$ from linear decomposition shown previously

Improving accuracy of baseline scheme

Baseline scheme: for $v \notin K$, π_v estimated as

$$\hat{\pi}_v = \sum_{k \in K} \mu_v(k) \pi_k$$

where $\mu_v(k)$ from linear decomposition shown previously

We show (for a certain function f)

$$\|\pi_v - \hat{\pi}_v\|_1 < \epsilon \Leftrightarrow \sum_{k \in K} \mu_v(k) > f(\epsilon)$$

Intuitively, small error $\Leftrightarrow$ v is "close" to K in graph

Improving accuracy of baseline scheme

Baseline scheme: for $v \notin K$, π_v estimated as

$$\hat{\pi}_v = \sum_{k \in K} \mu_v(k) \pi_k$$

where $\mu_v(k)$ from linear decomposition shown previously

We show (for a certain function f)

$$\|\pi_v - \hat{\pi}_v\|_1 < \epsilon \Leftrightarrow \sum_{k \in K} \mu_v(k) > f(\epsilon)$$

Intuitively, small error $\Leftrightarrow$ v is "close" to K in graph

Key point: $\sum_{k \in K} \mu_v(k)$ is (approximately) known at runtime!

$\Rightarrow$ If $\sum_{k \in K} \mu_v(k) < f(\epsilon)$, estimate π_v directly

Estimating PPR and linear combo weights (1/2)

Recall: π_v = stationary distribution of chain with transition matrix

$$P_v = \underbrace{(1-\alpha)P}_{\text{Random walk}} + \underbrace{\alpha 1_n e_v^T}_{\text{Jump to } v}$$

Estimating PPR and linear combo weights (1/2)

Recall: π_v = stationary distribution of chain with transition matrix

$$P_v = \underbrace{(1-\alpha)P}_{\text{Random walk}} + \underbrace{\alpha 1_n e_v^\mathsf{T}}_{\text{Jump to } v}$$

Solving $\pi_v = \pi_v P_v$ yields

$$\pi_v = \alpha e_v^\mathsf{T} (I_n - (1-\alpha)P)^{-1}$$

Estimating PPR and linear combo weights (1/2)

Recall: π_v = stationary distribution of chain with transition matrix

$$P_v = \underbrace{(1-\alpha)P}_{\text{Random walk}} + \underbrace{\alpha 1_n e_v^{\mathsf{T}}}_{\text{Jump to } v}$$

Solving $\pi_v = \pi_v P_v$ yields

$$\pi_v = \alpha e_v^{\mathsf{T}} (I_n - (1-\alpha)P)^{-1}$$

Since π_v is v-th row of Π,

$$\Pi = \alpha(I_n - (1-\alpha)P)^{-1} = \alpha \sum_{i=0}^{\infty} (1-\alpha)^i P^i$$

Estimating PPR and linear combo weights (1/2)

Recall: π_v = stationary distribution of chain with transition matrix

$$P_v = \underbrace{(1-\alpha)P}_{\text{Random walk}} + \underbrace{\alpha 1_n e_v^{\mathsf{T}}}_{\text{Jump to } v}$$

Solving $\pi_v = \pi_v P_v$ yields

$$\pi_v = \alpha e_v^{\mathsf{T}} (I_n - (1-\alpha)P)^{-1}$$

Since π_v is v-th row of Π,

$$\Pi = \alpha(I_n - (1-\alpha)P)^{-1} = \alpha \sum_{i=0}^{\infty} (1-\alpha)^i P^i$$

Suggests power iteration: choose i^* large and compute

$$\alpha \sum_{i=0}^{i^*} (1-\alpha)^i P^i \approx \Pi$$

Estimating PPR and linear combo weights (2/2)

Power iteration traverses all paths of length $\leq i^*$

Directed Laplacian variants:

- Set $i^* = \Theta(\log(n))$
- Modify power method so that dense matrices do not arise

Estimating PPR and linear combo weights (2/2)

Power iteration traverses all paths of length $\leq i^*$

Directed Laplacian variants:

- Set $i^* = \Theta(\log(n))$
- Modify power method so that dense matrices do not arise

Dynamic programming (DP) variants traverse only "important" paths

Estimating PPR and linear combo weights (2/2)

Power iteration traverses all paths of length $\leq i^*$

Directed Laplacian variants:

- Set $i^* = \Theta(\log(n))$
- Modify power method so that dense matrices do not arise

Dynamic programming (DP) variants traverse only "important" paths

Forward DP (Andersen, Chung, Lang 2006):

- Given v, traverses "important" paths out of v; estimates v-th row of Π
- **Can use to estimate PPR vectors directly**

Estimating PPR and linear combo weights (2/2)

Power iteration traverses all paths of length $\leq i^*$

Directed Laplacian variants:

- Set $i^* = \Theta(\log(n))$
- Modify power method so that dense matrices do not arise

Dynamic programming (DP) variants traverse only "important" paths

Forward DP (Andersen, Chung, Lang 2006):

- Given v, traverses "important" paths out of v; estimates v-th row of Π
- **Can use to estimate PPR vectors directly**

Backward DP (Andersen et al. 2008):

- Given v, traverses "important" paths into v; estimates v-th column of Π
- **Can use (modified version) to estimate linear combo weights**

Putting it all together

Our scheme estimates π_v . . .

- . . . by forward DP, if $v \in K$
- . . . by forward DP, if $v \notin K$ and linear combo determined to be inaccurate
- . . . as linear combo, if $v \notin K$ and linear combo determined to be accurate

Forward DP provably accurate; thus, all estimates are accurate

[4] Assuming $|E| = O(n), \alpha = \Theta(1/\log n)$

Putting it all together

Our scheme estimates π_v ...

- ... by forward DP, if $v \in K$
- ... by forward DP, if $v \notin K$ and linear combo determined to be inaccurate
- ... as linear combo, if $v \notin K$ and linear combo determined to be accurate

Forward DP provably accurate; thus, all estimates are accurate

Complexity dominated by number runs of forward DP

- By design, forward DP is run $\Delta(K, \epsilon)$ times
- Each run has $O(n \log n)$ complexity (by Andersen, Chung, Lang 2006)[4]

Overall complexity is $O(\Delta(K, \epsilon) n \log n) = o(n^2)$ (when theorem applies)

[4] Assuming $|E| = O(n), \alpha = \Theta(1/\log n)$

Comparison to existing algorithms

Best existing approach: run forward or backward DP $\forall\ v$

- l_1 accuracy guarantee, $O(n^2 \log n)$ complexity
- Ignores structure/dependencies across rows of Π!
- Our scheme accounts for structure, thus reduces complexity

Comparison to existing algorithms

Best existing approach: run forward or backward DP $\forall\ v$

- l_1 accuracy guarantee, $O(n^2 \log n)$ complexity
- Ignores structure/dependencies across rows of Π!
- Our scheme accounts for structure, thus reduces complexity

Another noteworthy work: Lofgren, Banerjee, Goel 2016

- Estimates single entry of Π via DP + MCMC, complexity $O(\sqrt{n} \log n)$
- Hence, $O(n^{2.5} \log n)$ to estimate Π; ignores dependencies across entries
- Again, accounting for structure allows us to reduce complexity

Comparison to existing algorithms

Best existing approach: run forward or backward DP $\forall\ v$

- l_1 accuracy guarantee, $O(n^2 \log n)$ complexity
- Ignores structure/dependencies across rows of Π!
- Our scheme accounts for structure, thus reduces complexity

Another noteworthy work: Lofgren, Banerjee, Goel 2016

- Estimates single entry of Π via DP + MCMC, complexity $O(\sqrt{n} \log n)$
- Hence, $O(n^{2.5} \log n)$ to estimate Π; ignores dependencies across entries
- Again, accounting for structure allows us to reduce complexity

Connections to other problems:

- Non-negative matrix factorization: Unknown $n \times n$ Π split into non-negative factors $n \times \tilde{k}$ and $\tilde{k} \times n$ factors in $o(n^2)$ time
  Related work Sen et al. 2016 is in a different norm.

Thanks!

Thanks for your attention

Paper appeared in ACM SIGMETRICS 2019

Questions?

References I

Andersen, Reid, Fan Chung, Kevin Lang (2006). "Local graph partitioning using PageRank vectors". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, pp. 475–486.

Andersen, Reid et al. (2008). "Local computation of PageRank contributions". In: *Internet Mathematics* 5.1-2, pp. 23–45.

Athreya, Krishna B, Örjan Stenflo (2003). "Perfect sampling for Doeblin chains". In: *Sankhyā: The Indian Journal of Statistics*, pp. 763–777.

Avrachenkov, Konstantin et al. (2007). "Monte Carlo methods in PageRank computation: When one iteration is sufficient". In: *SIAM Journal on Numerical Analysis* 45.2, pp. 890–904.

Baluja, Shumeet et al. (2008). "Video suggestion and discovery for YouTube: Taking random walks through the view graph". In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 895–904.

Bordenave, Charles, Pietro Caputo, Justin Salez (2018). "Random walk on sparse random digraphs". In: *Probability Theory and Related Fields* 170.3-4, pp. 933–960.

Chen, Ningyuan, Nelly Litvak, Mariana Olvera-Cravioto (2017). "Generalized PageRank on directed configuration networks". In: *Random Structures & Algorithms* 51.2, pp. 237–274.

Chen, Ningyuan, Mariana Olvera-Cravioto (2013). "Directed random graphs with given degree distributions". In: *Stochastic Systems* 3.1, pp. 147–186.

Freschi, Valerio (2007). "Protein function prediction from interaction networks using a random walk ranking algorithm". In: *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*. IEEE, pp. 42–48.

References II

Gupta, Pankaj et al. (2013). "WTF: The who to follow service at twitter". In: *Proceedings of the 22nd international conference on World Wide Web*. ACM, pp. 505–514.

Jeh, Glen, Jennifer Widom (2003). "Scaling personalized web search". In: *Proceedings of the 12th international conference on World Wide Web*. ACM, pp. 271–279.

Kloumann, Isabel M, Johan Ugander, Jon Kleinberg (2017). "Block models and personalized PageRank". In: *Proceedings of the National Academy of Sciences* 114.1, pp. 33–38.

Koutra, Danai, Joshua T Vogelstein, Christos Faloutsos (2013). "Deltacon: A principled massive-graph similarity function". In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, pp. 162–170.

Lofgren, Peter, Siddhartha Banerjee, Ashish Goel (2016). "Personalized PageRank estimation and search: A bidirectional approach". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, pp. 163–172.

Morrison, Julie L et al. (2005). "GeneRank: Using search engine technology for the analysis of microarray experiments". In: *BMC bioinformatics* 6.1, p. 233.

Page, Lawrence et al. (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab.

Sarma, Atish Das et al. (2015). "Fast distributed pagerank computation". In: *Theoretical Computer Science* 561, pp. 113–121.

Sen, Rajat et al. (2016). "Contextual bandits with latent confounders: An nmf approach". In: *arXiv preprint arXiv:1606.00119*.