Random-Radius Ball Method for Estimating Closeness Centrality

(appeared in *AAAI'17*)

稲荷場 渉 (東京大学)
秋葉 拓哉 (PFN)
吉田 悠一 (NII & PFI)

中心性 (Centrality)

複雑ネットワークの解析でしばしば気になるのが:

- *Q.* どの頂点がより中心的(重要)な役割を持っているか?
- *Q.* ある頂点の影響力は?

⟶ ***頂点の重要性*を測る指標を中心性と呼ぶ**

[https://twitter.com/katyperry]

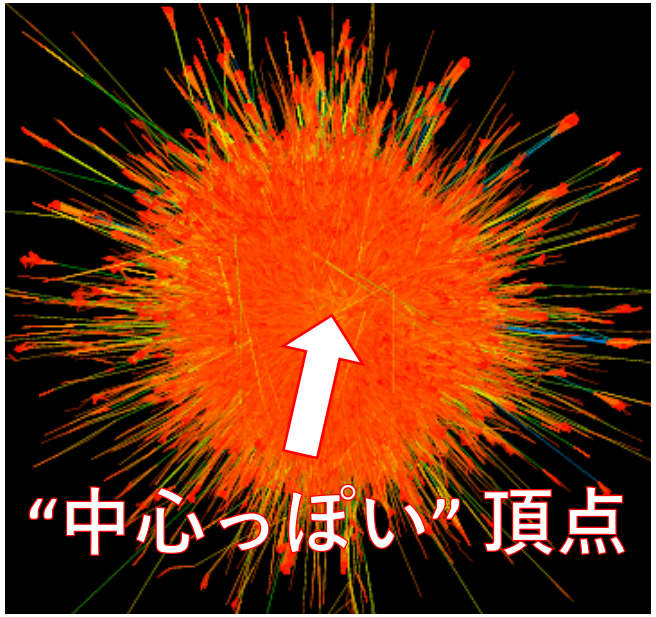
中心性 (Centrality)

複雑ネットワークの解析でしばしば気になるのが:

- ***Q.*** どの頂点がより中心的(重要)な役割を持っているか?
- ***Q.*** ある頂点の影響力は?

⟶ ***頂点の重要性*を測る指標を中心性と呼ぶ**

- Classic closeness centrality
  [Bavelas *J. Acoust. Soc. Am. '50*]
- Kats centrality [Kats *Psychometrika '54*]
- Betweenness centrality
  [Anthonisse '*71*] [Freeman *Sociometry '77*]
- PageRank
  [Brin-Page *Comput. Networks ISDN '98*]
- Harmonic centrality
  [Opsahl-Agneessens-Skvoretz *Soc. Networks '10*]
  [Boldi-Vigna *Internet Math. '14*]



[http://www.cise.ufl.edu/research/sparse/matrices/SNAP/soc-LiveJournal1.html]

中心性 (Centrality)

複雑ネットワークの解析でしばしば気になるのが:

- ***Q.*** どの頂点がより中心的(重要)な役割を持っているか?
- ***Q.*** ある頂点の影響力は?

⟹ ***頂点の重要性*を測る指標を中心性と呼ぶ**

- Classic closeness centrality
  [Bavelas *J. Acoust. Soc. Am. '50*]
- Kats centrality
  [Kats *Psychometrika '54*]
- Betweenness centrality
  [Anthonisse *'71*] [Freeman *Sociometry '77*]
- PageRank
  [Brin-Page *Comput. Networks ISDN '98*]
- Harmonic centrality
  [Opsahl-Agneessens-Skvoretz *Soc. Networks '10*]
  [Boldi-Vigna *Internet Math. '14*]

$$C_{cc}(v) := \frac{1}{\sum_u d(u,v)}$$

他のすべての頂点からの距離の総和

(強)連結でないグラフの扱い…?

$d(u,v) :=$ **u から v への距離**

中心性 (Centrality)

複雑ネットワークの解析でしばしば気になるのが:

Q. どの頂点がより中心的(重要)な役割を持っているか?

Q. ある頂点の影響力は?

→ ***頂点の重要性*を測る指標を中心性と呼ぶ**

- Classic closeness centrality
  [Bavelas *J. Acoust. Soc. Am. '50*]
- Kats centrality
  [Kats *Psychometrika '54*]
- Betweenness centrality
  [Anthonisse *'71*] [Freeman *Sociometry '77*]
- PageRank
  [Brin-Page *Comput. Networks ISDN '98*]
- **Harmonic centrality**
  [Opsahl-Agneessens-Skvoretz *Soc. Networks '10*]
  [Boldi-Vigna *Internet Math. '14*]

$$C_{cc}(v) := \frac{1}{\sum_u d(u,v)}$$

より洗練された定義

$$C(v) := \sum_u \frac{1}{d(u,v)}$$

$d(u,v) := u$ から v への距離

問題定義

調和中心性 (Harmonic centrality): $C(v) := \sum_u \frac{1}{d(u,v)}$

Goal: 全頂点の $C(v)$ の計算

全頂点の $C(v)$ が得られれば，次は容易に計算可能

- 頂点ランキングの作成 (または top-k 頂点の検出)
- 各頂点に対する中心性クエリへの応答

ところが… ナイーブな実装は $O(nm)$-時間 を要する

- 今日の巨大ネットワークでは動作しない

n: 頂点数，m: 辺数

⟹ **高速な近似アルゴリズムが求められる**

貢献

Random-Radius Ball (RRB) 法

- 調和中心性に対する新たな近似アルゴリズムを提案

RRB法の基本部

精度パラメータを
明示的に指定できない

ブートストラップ

指定の精度を満たすまで
RRB法の基本部を繰り返す

- 理論的解析: 計算量および近似に関する誤差保証
- 実データを用いた実験

$$C_\alpha(v) := \sum_u \alpha\big(d(u,v)\big)$$

$\alpha(x) \geq 0$ かつ $\alpha(\infty) = 0$ を
満たす単調減少関数

調和中心性以外の指標にも適用可能:

- $\alpha(x) = 1/x$: 調和中心性
- $\alpha(1) = 1,\ \alpha(2) = 0$: 頂点の次数

既存手法との比較

調和中心性の推定が行える既存手法は2つ:

HyperBall [Boldi-Vigna *ICDMW '13*]

- 実験的に高速 / HyperLogLog データ構造を利用
  [Flajolet et al. *AofA '07*]

All-Distances Sketches (ADS) [Cohen *TKDE '15*]

- さまざまな指標を理論保証付きで推定可能

	低い メモリ使用量	誤差に関する 理論保証	実装の容易さ
RRB法 (提案手法)	✓	✓	✓
HyperBall	✓	-	-
All-Distances Sketches	-	✓	✓

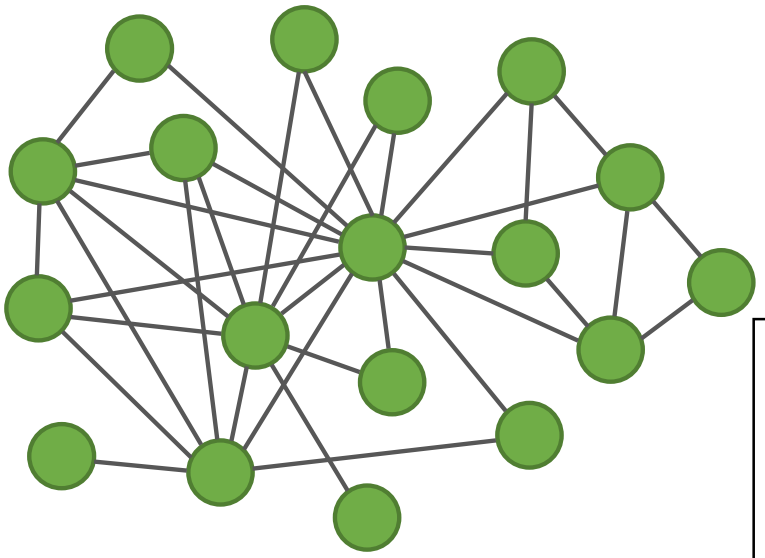
Random-Radius Ball (RRB) 法

入力: グラフ $G = (V, E)$, パラメータ $0 < t \leq 1$.

foreach vertex $u \in V$ **do**

$\tau \leftarrow \left\lfloor \frac{t}{\mathrm{rand}()} \right\rfloor$. (rand() は $(0, 1]$ の一様な乱数を返すものとする)

頂点 u から深さ τ の BFS を行い，訪れた頂点をマーク.



u からの BFS が v を訪れる確率が $\frac{t}{d(u,v)}$ となるように設定

- $\mathrm{rand}() \in (0, \frac{t}{d(u,v)}]$ のとき $\tau \geq d(u, v)$.

なぜBFSを途中で打ち切るのか?

近い頂点の個数を正確に推定したい

遠い頂点の個数の見積もりは雑でも良い

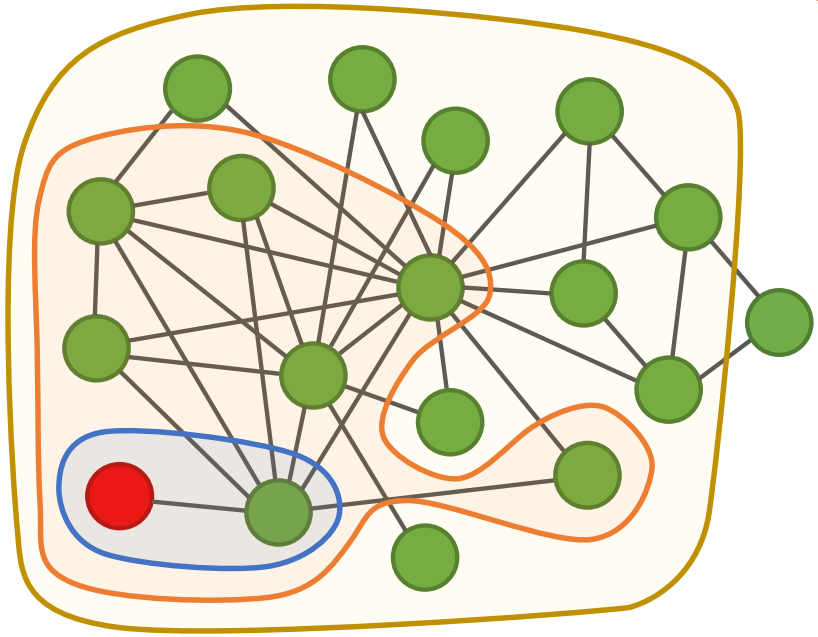
Random-Radius Ball (RRB) 法

入力: グラフ $G = (V, E)$, パラメータ $0 < t \leq 1$.

foreach vertex $u \in V$ **do**

$\tau \leftarrow \left\lfloor \frac{t}{\mathrm{rand}()} \right\rfloor$. (rand() は $(0, 1]$ の一様な乱数を返すものとする)

頂点 u から深さ τ の BFS を行い，訪れた頂点をマーク.



u からの BFS が v を訪れる確率が $\frac{t}{d(u,v)}$ となるように設定

例:

左下の赤い頂点からのBFSは

- 確率 t で青色の領域に到達
- 確率 $t/2$ で橙色の領域に到達
- 確率 $t/3$ で黄色の領域に到達

Random-Radius Ball (RRB) 法

入力: グラフ $G = (V, E)$, パラメータ $0 < t \leq 1$.

foreach vertex $u \in V$ **do**

$\tau \leftarrow \left\lfloor \frac{t}{\mathrm{rand}()} \right\rfloor$. (rand() は $(0, 1]$ の一様な乱数を返すものとする)

頂点 u から深さ τ の BFS を行い，訪れた頂点をマーク.

最終的に，ある頂点 v がマークされた回数は?

$$\mathbf{E}[v \text{ のマーク回数}] = \sum_u \frac{t}{d(u,v)} = t \cdot C(v)$$

- $\hat{C}(v) := \frac{(v \text{ のマーク回数})}{t}$ が $C(v)$ の不偏推定量となる!

RRB法の基本部: 解析

C_{avg} を中心性の平均値，また $k_t := t \cdot C_{avg}$ と定める．

計算量:

- RRB法は $O(n)$ の (追加の) 空間計算量を要する．
- 時間計算量は $O(Rk_t m)$，ただし $R = \dfrac{\max C(v)}{C_{avg}}$．

グラフがスモールワールドなら
R は定数

誤差保証:

- 頂点 v が $C(v) \geq C_{avg}$ を満たすものとする．このとき，

相対誤差の標準偏差
(変動係数; CV)

$$= \frac{\sigma\left(\hat{C}(v)\right)}{C(v)} \leq \frac{1}{\sqrt{k_t}}.$$

RRB法の基本部: 解析

C_{avg}を中心性の平均値, また $k_t := t \cdot C_{avg}$ と定める.

値は一般に分からない ⟹ k_t が指定できない!

計算量:

- RRB法は $O(n)$ の (追加の) 空間計算量を要する.
- 時間計算量は $O(Rk_t m)$, ただし $R = \frac{\max C(v)}{C_{avg}}$.

パラメータ k_t が
計算量と精度の両方を定める

誤差保証:

- 頂点 v が $C(v) \geq C_{avg}$ を満たすものとする. このとき,

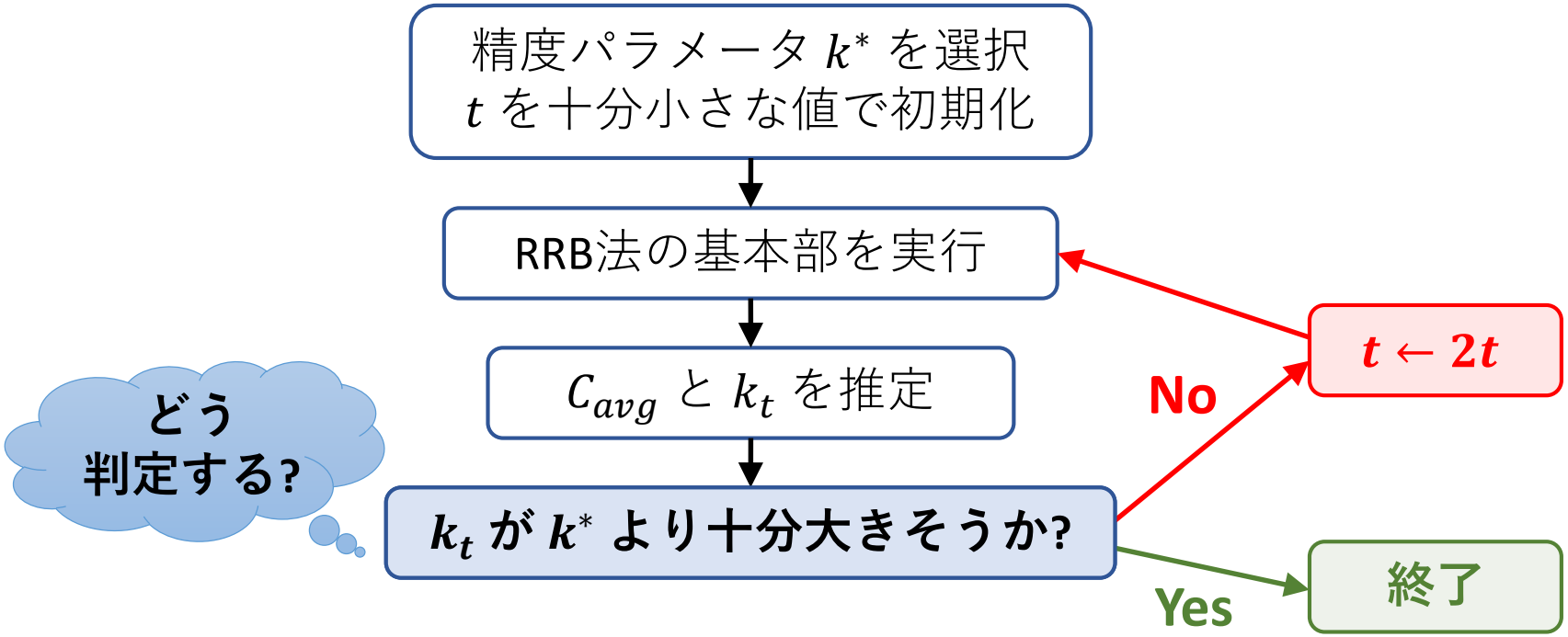
相対誤差の標準偏差 (変動係数; CV) $= \frac{\sigma\left(\hat{C}(v)\right)}{C(v)} \leq \frac{1}{\sqrt{k_t}}$.

“ブートストラップ” 法

問題: パラメータ k_t を制御する方法はあるか?

解決策: RRB法を実行した結果を基に k_t を “推定” する

“ブートストラップ” の枠組み



“ブートストラップ” 法: 解析

k_t の推定値

基準: $\widehat{k_t} \geq k^* + s\sqrt{k^*}$ のとき，k_t が k^* より大きそうと判断

ブートストラップ法のパラメータ

時間計算量:

- $O\left(R\left(k^* + (2+s)\sqrt{k^*}\right)m\right)$-時間 で動作. (Thm. 10)

時間計算量は
ほぼ増加しない

誤差保証:

- 頂点 v が $C(v) \geq C_{avg}$ を満たすものとする．このとき，

$$\frac{\sigma\left(\hat{C}(v)\right)}{C(v)} \leq \frac{1}{\sqrt{k^*}}$$

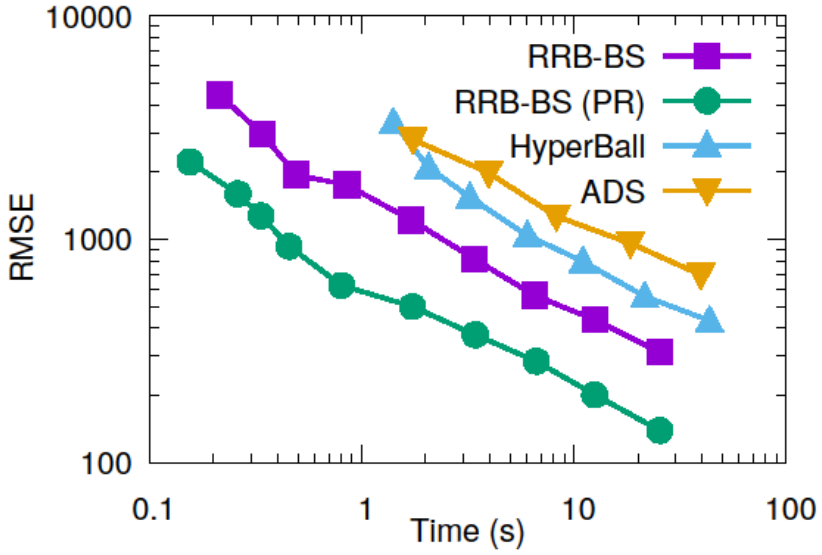
が **確率 $1 - 4/k^* - 1/s^2$ 以上で** 成り立つ. (Thm. 9)

わずかなオーバーヘッドで精度の制御が可能に

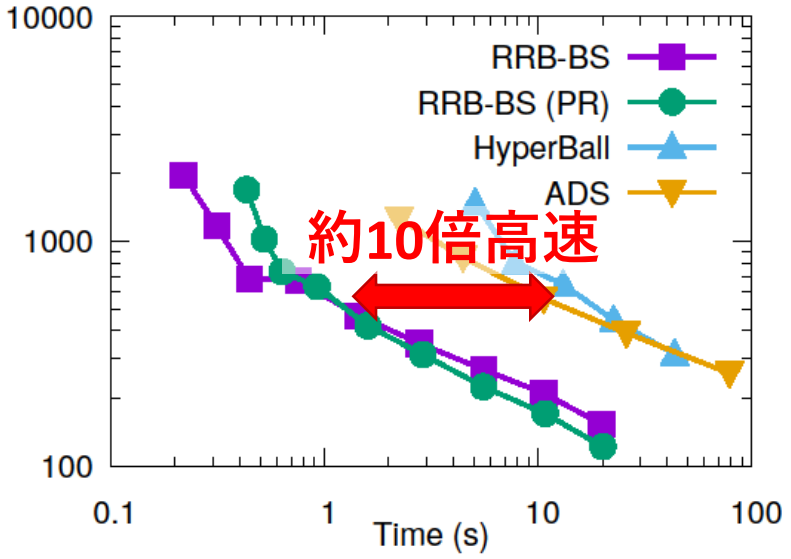
実験: 既存手法との比較

計算時間と精度のトレードオフ

soc-Slashdot0902 (Social)
n = 82,168 m = 948,464



web-NotreDame (Web)
n = 325,729 m = 1,497,134



- RRB法は既存手法より優れた性能
- Permutation-rank 法 (PR) を用いると実験的な精度は改善 (理論保証は無し)

(Intel Xeon E5540 processors (2.53GHz), 48GiB RAM, single-threaded)

実験: スケーラビリティ

ブートストラップ法 $(k^* = 100,\ s = 3)$

データセット	種別	頂点数	辺数	時間 (秒)	誤差
web-Google	Web	876 K	5.1 M	24.69	3.90%
com-youtube	Social	1.1 M	6.0 M	20.99	2.38%
dblp-2001	Social	933 K	6.7 M	29.65	2.92%
ego-Gplus	Social	108 K	14 M	10.10	4.83%
in-2004	Web	1.4 M	17 M	27.49	5.84%
soc-Pokec	Social	1.6 M	31 M	107.93	3.67%
soc-LiveJournal1	Social	4.8 M	69 M	278.03	2.54%
enwiki-2013	Social	4.2 M	101 M	223.64	1.72%

- 1億辺規模のネットワークも取り扱える
- 理論保証の 10% $(= 1/\sqrt{100})$ よりも実際の誤差は小さい

(Intel Xeon E5540 processors (2.53GHz), 48GiB RAM, single-threaded)

実験: スケーラビリティ

ブートストラップ法 $(k^* = 100,\ s = 3)$

データセット	種別	頂点数	辺数	時間 (秒)	誤差
web-Google	Web	876 K	5.1 M	24.69	3.90%
com-youtube	Social	1.1 M	6.0 M	20.99	2.38%
dblp-2001	Social	933 K	6.7 M	29.65	2.92%
ego-Gplus	Social	108 K	14 M	10.10	4.83%
in-2004	Web	1.4 M	17 M	27.49	5.84%
soc-Pokec	Social	1.6 M	31 M	107.93	3.67%
soc-LiveJournal1	Social	4.8 M	69 M	278.03	2.54%
enwiki-2013	**Social**	**4.2 M**	**101 M**	**223.64**	**1.72%**

- 1億辺規模のネットワークも取り扱える
- 理論保証の 10% $(= 1/\sqrt{100})$ よりも実際の誤差は小さい

(Intel Xeon E5540 processors (2.53GHz), 48GiB RAM, single-threaded)

厳密な計算には
24スレッド×2日半を
要した

まとめ

距離に基づく中心性の推定が高速に行える
random-radius ball (RRB) 法 を提案

- 特に頂点ランキングの上位を求めるような応用に適する

今後の可能性

- RRB法の応用範囲は中心性に限らない:
  少しの変更で ***closeness similarity*** (近接類似性) も推定可能
  [Cohen et al. *COSN '13*]

詳細な比較　k, s: 計算量と精度のトレードオフパラメータ

	時間計算量	空間	誤差保証 (CV)
RRB法	$O\left(\left(k+(2+s)\sqrt{k}\right)m \cdot \frac{\max C(v)}{C_{avg}}\right)$	$O(n)$	$\leq 1/\sqrt{k}$ $\forall v$ s.t. $C(v) \geq C_{avg}$ with prob. $1-4/k-1/s^2$
HyperBall	$O\left(km \cdot \mathrm{diam}(G)\right)$	$O(kn)$	$\leq O\left(1/\sqrt{k}\right)$ ただし実験的にのみ
ADS	$O(k(m+n\log k)\log n)$	$O(kn)$	$\leq 1/\sqrt{2(k-1)}$