Important problems of graph theory and the Internet

Andrei Raigorodskii

Lomonosov Moscow State University,
Moscow Institute of Physics and Technology,
Yandex Division of Theoretical and Applied Research,
Moscow, Russia

GraphHPC-2014, 04 March 2014

The main objects

Real-world web-graph

$G = (V, E)$, where V —

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).
Sometimes multiple edges are identified. Sometimes multiple edges and even loops are allowed.

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).
Sometimes multiple edges are identified. Sometimes multiple edges and even loops are allowed.

How can we use this graph for practical purposes?

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).
Sometimes multiple edges are identified. Sometimes multiple edges and even loops are allowed.

How can we use this graph for practical purposes?

- Use its structure to find features improving learning to rank;

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).
Sometimes multiple edges are identified. Sometimes multiple edges and even loops are allowed.

How can we use this graph for practical purposes?

- Use its structure to find features improving learning to rank;
- Adjust algorithms including, say, those for crawling;

The main objects

Real-world web-graph

$G = (V, E)$, where V —

- set of web-pages,
- set of web-sites,
- set of web-hosts,

and E — the set of all hyperlinks between the vertices (nodes).
Sometimes multiple edges are identified. Sometimes multiple edges and even loops are allowed.

How can we use this graph for practical purposes?

- Use its structure to find features improving learning to rank;
- Adjust algorithms including, say, those for crawling;
- Find unexpected structures such as news, spam, etc.

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.
- Web-graphs have a unique "giant" connected component.

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.
- Web-graphs have a unique "giant" connected component.
- Every two vertices in the giant component are connected by a path of short length (5–6, 15–20 depending on what we mean by web-graph): $\operatorname{diam} G \approx 6$ (the rule of 6 handshakes).

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.
- Web-graphs have a unique "giant" connected component.
- Every two vertices in the giant component are connected by a path of short length (5–6, 15–20 depending on what we mean by web-graph): $\operatorname{diam} G \approx 6$ (the rule of 6 handshakes).
- Web-graphs are robust when random vertices are destroyed (a giant component survives).

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.
- Web-graphs have a unique "giant" connected component.
- Every two vertices in the giant component are connected by a path of short length (5–6, 15–20 depending on what we mean by web-graph): $\operatorname{diam} G \approx 6$ (the rule of 6 handshakes).
- Web-graphs are robust when random vertices are destroyed (a giant component survives).
- Web-graphs are vulnerable to attacks onto hubs (many small components appear after a threshold is surpassed).

Some important properties/features, which must be fastly checked on real data

Barabási–Albert, Watts–Strogatz, Newman, and many others in 90s–00s.

- Web-graphs are *sparse*, i.e., their numbers of edges (links) are proportional to their numbers of vertices.
- Web-graphs have a unique "giant" connected component.
- Every two vertices in the giant component are connected by a path of short length (5–6, 15–20 depending on what we mean by web-graph): $\operatorname{diam} G \approx 6$ (the rule of 6 handshakes).
- Web-graphs are robust when random vertices are destroyed (a giant component survives).
- Web-graphs are vulnerable to attacks onto hubs (many small components appear after a threshold is surpassed).
- The degree distribution is close to a power-law:

$$\frac{|\{v \in V : \ \deg v = d\}|}{n} \sim \frac{const}{d^\gamma},$$

where $\gamma \in (2, 3)$ depends on what we mean by web-graph.

More of important properties/features: global clustering

More of important properties/features: global clustering

Let $G = (V, E)$ be a simple graph, $|V| = n$. Let $v \in V$. Denote by N_v the set of neighbours of v in G. Set $n_v = |N_v|$, i.e., $n_v = \deg v$. If $n_v \geqslant 2$, then *the clustering coefficient* of the vertex v is

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

More of important properties/features: global clustering

Let $G = (V, E)$ be a simple graph, $|V| = n$. Let $v \in V$. Denote by N_v the set of neighbours of v in G. Set $n_v = |N_v|$, i.e., $n_v = \deg v$. If $n_v \geqslant 2$, then *the clustering coefficient* of the vertex v is

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 1

The global clustering coefficient of G is

$$T(G) = \frac{\sum\limits_{v \in V} C_{n_v}^2 C_v}{\sum\limits_{v \in V} C_{n_v}^2}.$$

More on global clustering

Let $\sharp(H, G)$ be the number of copies of a graph H in a graph G.

More on global clustering

Let $\sharp(H, G)$ be the number of copies of a graph H in a graph G.

Clearly

$$T(G) = \frac{\sum\limits_{v \in V} C_{n_v}^2 C_v}{\sum\limits_{v \in V} C_{n_v}^2} = \frac{3\sharp(K_3, G)}{\sharp(P_2, G)},$$

where K_3 is a triangle and P_2 is a 2-path.

More on global clustering

Let $\sharp(H, G)$ be the number of copies of a graph H in a graph G.

Clearly

$$T(G) = \frac{\sum\limits_{v \in V} C_{n_v}^2 C_v}{\sum\limits_{v \in V} C_{n_v}^2} = \frac{3\sharp(K_3, G)}{\sharp(P_2, G)},$$

where K_3 is a triangle and P_2 is a 2-path.

Thus, roughly speaking, $T(G)$ is the probability that two neighbours of a vertex of G are themselves joined by an edge.

More on global clustering

Let $\sharp(H, G)$ be the number of copies of a graph H in a graph G.

Clearly

$$T(G) = \frac{\sum\limits_{v \in V} C_{n_v}^2 C_v}{\sum\limits_{v \in V} C_{n_v}^2} = \frac{3\sharp(K_3, G)}{\sharp(P_2, G)},$$

where K_3 is a triangle and P_2 is a 2-path.

Thus, roughly speaking, $T(G)$ is the probability that two neighbours of a vertex of G are themselves joined by an edge.

Clearly the last formula for $T(G)$ may be used for any graph, not only a simple one.

More of important properties/features: local clustering

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

- As a rule, $T(G) < C(G)$.

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

- As a rule, $T(G) < C(G)$.
- Both values are constant in real-world graphs (high clustering).

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

- As a rule, $T(G) < C(G)$.
- Both values are constant in real-world graphs (high clustering).
- $C(G)$ is much easier to calculate.

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x, y\} \in E : \ x, y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

- As a rule, $T(G) < C(G)$.
- Both values are constant in real-world graphs (high clustering).
- $C(G)$ is much easier to calculate.
- $C(G)$ cannot be naturally defined for graphs with multiple edges and loops.

More of important properties/features: local clustering

$$C_v = \frac{|\{\{x,y\} \in E : \ x,y \in N_v\}|}{C_{n_v}^2}.$$

Clustering coefficient 2

The local clustering coefficient of G is

$$C(G) = \frac{1}{n} \sum_{v \in V} C_v.$$

The values $T(G)$ and $C(G)$ are rather different.

- As a rule, $T(G) < C(G)$.
- Both values are constant in real-world graphs (high clustering).
- $C(G)$ is much easier to calculate.
- $C(G)$ cannot be naturally defined for graphs with multiple edges and loops.
- $C(G)$ is more complicated for theoretical study (in random graphs).

Degree correlations

Degree correlations

Assortativity

For a simple graph, let

$$d_{nn}(d) = \frac{1}{d \cdot |\{i : \ \deg i = d\}|} \sum_{i:\ \deg\ i=d} \sum_{j:\ \{i,j\}\in E} \deg j.$$

Degree correlations

Assortativity

For a simple graph, let

$$d_{nn}(d) = \frac{1}{d \cdot |\{i : \ \deg i = d\}|} \sum_{i:\ \deg\ i=d} \sum_{j:\ \{i,j\}\in E} \deg j.$$

As a rule, $d_{nn}(d) \sim d^{\delta}$.

Degree correlations

Assortativity

For a simple graph, let

$$d_{nn}(d) = \frac{1}{d \cdot |\{i : \ \deg i = d\}|} \sum_{i:\ \deg\ i=d} \sum_{j:\ \{i,j\}\in E} \deg j.$$

As a rule, $d_{nn}(d) \sim d^{\delta}$.

- For web-graphs, $\delta < 0$ (called *dissassortative network*).

Degree correlations

Assortativity

For a simple graph, let

$$d_{nn}(d) = \frac{1}{d \cdot |\{i : \ \deg i = d\}|} \sum_{i:\ \deg\ i=d} \sum_{j:\ \{i,j\}\in E} \deg j.$$

As a rule, $d_{nn}(d) \sim d^{\delta}$.

- For web-graphs, $\delta < 0$ (called *dissassortative network*).
- For social networks, $\delta > 0$ (called *assortative networks*).

Page Ranks

Page Ranks

Classical PageRank

Let $G_n = (V_n, E_n)$ be a web-graph. Denote by $PR(v)$ a PageRank of a vertex v. Define $PR(i)$, where $i \in \{1, \ldots, |V_n|\}$, as the solution of the following system of linear equations:

$$PR(i) = c \sum_{j \to i} \frac{PR(j)}{\text{outdeg } j} + \frac{c}{|V_n|} \sum_{j \in \mathcal{D}} PR(j) + \frac{1-c}{|V_n|}, \quad i = 1, \ldots, |V_n|,$$

where $c \in (0, 1)$ is a constant and $\mathcal{D}$ is the set of vertices having zero outdegrees.

Page Ranks

Classical PageRank

Let $G_n = (V_n, E_n)$ be a web-graph. Denote by $PR(v)$ a PageRank of a vertex v. Define $PR(i)$, where $i \in \{1, \ldots, |V_n|\}$, as the solution of the following system of linear equations:

$$PR(i) = c \sum_{j \to i} \frac{PR(j)}{\text{outdeg } j} + \frac{c}{|V_n|} \sum_{j \in \mathcal{D}} PR(j) + \frac{1-c}{|V_n|}, \quad i = 1, \ldots, |V_n|,$$

where $c \in (0, 1)$ is a constant and $\mathcal{D}$ is the set of vertices having zero outdegrees.

Very complicated calculations and many algorithms for approximation.

Page Ranks

Classical PageRank

Let $G_n = (V_n, E_n)$ be a web-graph. Denote by $PR(v)$ a PageRank of a vertex v. Define $PR(i)$, where $i \in \{1, \ldots, |V_n|\}$, as the solution of the following system of linear equations:

$$PR(i) = c \sum_{j \to i} \frac{PR(j)}{\text{outdeg } j} + \frac{c}{|V_n|} \sum_{j \in \mathcal{D}} PR(j) + \frac{1-c}{|V_n|}, \quad i = 1, \ldots, |V_n|,$$

where $c \in (0, 1)$ is a constant and $\mathcal{D}$ is the set of vertices having zero outdegrees.

Very complicated calculations and many algorithms for approximation.

However, now Classical PR is not a strong feature!

Page Ranks

Classical PageRank

Let $G_n = (V_n, E_n)$ be a web-graph. Denote by $PR(v)$ a PageRank of a vertex v. Define $PR(i)$, where $i \in \{1, \ldots, |V_n|\}$, as the solution of the following system of linear equations:

$$PR(i) = c \sum_{j \to i} \frac{PR(j)}{\text{outdeg } j} + \frac{c}{|V_n|} \sum_{j \in \mathcal{D}} PR(j) + \frac{1-c}{|V_n|}, \quad i = 1, \ldots, |V_n|,$$

where $c \in (0, 1)$ is a constant and $\mathcal{D}$ is the set of vertices having zero outdegrees.

Very complicated calculations and many algorithms for approximation.

However, now Classical PR is not a strong feature!

So many even more sophisticated algorithms.