Dynamics and security of network navigation

Dusko Pavlovic
Nov 2006
Motivation: Web security

Security is a social concept

• intuitively:
  – "Alice", "Bob", "speaks for", "believes"

• technically:
  – Web computation is a social process
Computer 1950

\[
\text{state} \xrightarrow{\text{transition}} \text{state}
\]
Computer 1950

state \xrightarrow{transition} state

computation (language)
Computer 1950

transition

state ⟷ state

computation (language)

effects
Computer 1990

transition

state -> state

computation
(language)

effects
Network 2000

traffic
(routing, surfing)

effects
Difference

effects of computation: state changes

effects of network traffic: social, economic

• humans as computational agents
  – hidden interactions, entanglement

• information flow is marketing
  – Overture, Google…
  – new dynamics of branding
adversarial interests: Tragedy of the Commons

- it is rational to exhaust common resources
  - otherwise the neighbors will do it

- it is not rational to invest in protection
  - because the neighbors will not

insecure traffic: Web-specific threats

- pharming, phishing, identity theft

- spamdexing, click fraud, impression fraud, splogging
(Prisoners’ Dilemma of security)

<table>
<thead>
<tr>
<th></th>
<th>$c$</th>
<th>$p\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>$c + q\ell$</td>
<td>$p\ell + \bar{p}q\ell$</td>
</tr>
<tr>
<td>$c + q\ell$</td>
<td>$p\ell + \bar{p}q\ell$</td>
<td></td>
</tr>
<tr>
<td>$p\ell$</td>
<td>$p\ell + \bar{p}q\ell$</td>
<td></td>
</tr>
</tbody>
</table>

where

- $c$ — cost of protection
- $\ell$ — cost of infection
- $p$ — risk of infection if unprotected
- $q$ — risk of infection from neighbor
Task

Model Web as information source

• develop tools to analyze its dynamics
  – attacks have statistical profile

• map its structure: communities, cliques...
  – attackers have statistical profile

• measure authenticity information-theoretically
  – like in human communication
Background

theory of complex networks: random graphs, scaling, power laws, HOT...

- derives structure from evolution

link surfing models: PageRank, HITS, SALSA...

- Web search and information retrieval
In order to

- authenticate Web flows

- detect attacker clusters

need to model

- not just one-hop link surfing
  - keyword search

- but also path navigation
  - concept association
1. Introduction

2. Survey of link surfing and search

3. Path navigation

4. Communities, cliques and clusters

5. Rank distribution and optimal coding

6. Future work
Link network is a directed graph

\[ A = \begin{pmatrix} \delta \\ \rho \end{pmatrix} (E \xrightarrow{\delta} N) \]

which can be viewed as a matrix with the entries

\[ A_{ij} = \left\{ e \in E \mid \delta(e) = i \land \rho(e) = j \right\} \]

\[ = \left\{ e \in E \mid i \xrightarrow{e} j \right\} \]
Network dynamics

forward: probability that surfer at $i$ goes to $j$

$$A_{ij}^> = \frac{A_{ij}}{A_{i\cdot}}$$  where

$$A_{i\cdot} = \sum_{k=1}^{N} A_{ik}$$

backward: probability that surfer at $j$ comes from $i$

$$A_{ij}^< = \frac{A_{ij}}{A_{\cdot j}}$$  where

$$A_{\cdot j} = \sum_{k=1}^{N} A_{kj}$$
**Node ranking**

**push rank (promotion):** probability that a surfer departs from $i$

$$r_i^p = \sum_{k=1}^{N} A_{ik} r_k^p$$

**pull rank (reputation):** probability that a surfer arrives at $j$

$$r_j^s = \sum_{k=1}^{N} r_k^s A_{kj}$$
Reputation (\(\sim\) PageRank)
Influence

\[ i \rightarrow j \]
Node ranking (bis)

push rank (promotion):

\[ r^\prec = A^\prec r^\prec \]
\[ r^\prec_i = \text{Prob}(\delta(\alpha) = i \mid \alpha \in \text{paths}(E)) \]

pull rank (reputation):

\[ r^\succ = r^\succ A^\succ \]
\[ r^\succ_j = \text{Prob}(q(\alpha) = j \mid \alpha \in \text{paths}(E)) \]
Expected flow

\[
    r_{i\ell}^\diamond = r_i^\triangleleft r_\ell^\triangleright
\]

\[
    r_{i\ell}^\circ = \sum_{j=1}^{N} \sum_{k=1}^{N} A_{ij}^\triangleleft r_j^\diamond r_k^\triangledown A_{k\ell}^\triangleright
\]
1. Introduction

2. Survey of link surfing and search

3. **Path navigation**

4. Communities, cliques and clusters

5. Rank distribution and optimal coding

6. Future work
Paths in a network

\[(i = i_0) \xrightarrow{a_1} i_1 \xrightarrow{a_2} i_2 \rightarrow \cdots \xrightarrow{a_n} (i_n = j)\]

Motivation. Impose equations to hide irrelevant details

- identify all paths \(i \rightarrow j\) of the same length (same cost/payoff)

- identify all paths \(i \rightarrow j\) which go through \(k \xrightarrow{e} \ell\)

- count the visitors from a particular set of nodes, ignore their intermittent behavior
Path network

\[ A = (E \Rightarrow N) \]

comes equipped with sequential composition

\[ A_{ij} \times A_{jk} \rightarrow A_{ik} \]

\[ \langle i \xrightarrow{\alpha} j, j \xrightarrow{\beta} k \rangle \leftrightarrow (i \xrightarrow{\alpha \cdot \beta} k) \]

which is associative and unitary

\[ \alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma \]

\[ \gamma \cdot \iota_\ell = \gamma = \iota_i \cdot \gamma \]
Network of path extensions

Given a path network $A$, form

$$\tilde{A} = (\tilde{E} \xrightarrow{\delta} \tilde{N})$$

where

$$\tilde{N} = E$$
$$\tilde{E} = \sum_{\alpha, \beta \in E} \tilde{A}_{\alpha\beta}$$

where

$$\tilde{A}_{\alpha\beta} = \left\{ \langle \varphi_0, \varphi_1 \rangle \in E \times E \mid \begin{array}{c}
\varphi_0 \\
\alpha \\
\downarrow \\
\downarrow \\
j
\varphi_1 \\
\beta \\
\downarrow \\
\downarrow \\
k
\ell
\end{array} \right\}$$

25
Dynamics of path extensions

**forward:** probability that a random extension of $\alpha$ is $\beta$

$$\tilde{A}_\alpha^\uparrow \beta = \frac{\tilde{A}_{\alpha \beta}}{\tilde{A}_\alpha}$$

**backward:** probability that a random restriction of $\beta$ is $\alpha$

$$\tilde{A}_\alpha^\downarrow \beta = \frac{\tilde{A}_{\alpha \beta}}{\tilde{A}_\cdot \beta}$$
Path ranking

**push rank (extensibility):** probability that $\alpha$ is extended

$$\bar{r}_{\alpha}^{<} = \sum_{\beta \in \tilde{N}} \bar{A}_{\alpha\beta}^{<} \bar{r}_{\beta}^{<}$$

**pull rank (attraction):** probability that $\beta$ is traversed

$$\bar{r}_{\beta}^{>} = \sum_{\alpha \in \tilde{N}} \bar{r}_{\alpha}^{>} \bar{A}_{\alpha\beta}^{>}$$
**Proposition.** Total attraction of the paths between two nodes is equal to the expected flow:

\[
\sum_{i \rightarrow \ell} r_i^\beta = r_{i\ell}^\circ
\]

**Corollary.** Promotion and reputation are the marginals of total attraction:

\[
\sum_{i \rightarrow \bullet} r_i^\beta = r_i^\triangleleft
\]

\[
\sum_{\bullet \rightarrow \ell} r_\ell^\beta = r_\ell^\triangleright
\]
Interpretation

Proposition implies that

\[
\text{Prob}(\xi = \beta \mid i \xrightarrow{\xi} \ell \in E) = \frac{r_\beta}{r_{i\ell}}
\]
Path bias

\[ \Upsilon_{\beta} = \bar{r}_{\beta} - \frac{r_{i\ell}^\circ}{A_{i\ell}} \]

**it does** measure the bias of the paths between two fixed nodes

- i.e., how much the flow through a path deviates from the expected flow (resulting from the push away from the start of the path, and from the pull to its end);

**it does not** detect nodes with unexpected flows
Outline

1. Introduction

2. Survey of link surfing and search

3. Path navigation

4. Communities, cliques and clusters

5. Rank distribution and optimal coding

6. Future work
Contracts

are represented in a path network $A$ by a distinguished family of surfing (computation) paths

$$C = \{ j \xrightarrow{\alpha} k \in E \}$$

Motivation.

- advertising contracts
- buy-sell relationships
- links (or paths shorter than $n$ hops)
- paths with positive cost/profit ratio
Contract network

Given a network $A$ with a family of contracts $C$, form

$$\tilde{A} = (\tilde{E} \xrightarrow{\delta} \tilde{N})$$

where

$$\tilde{N} = C$$
$$\tilde{E} = \sum_{\alpha, \beta \in E} \tilde{A}_{\alpha \beta} \text{ where}$$

$$\tilde{A}_{\alpha \beta} = \left\{ \langle \varphi_0, \varphi_1 \rangle \in E \times E \mid \begin{array}{c} j \varphi_0 \beta \alpha \varphi_1 \kappa \ell \end{array} \right\}$$
Contract path

![Diagram](image.png)

can be thought of as contract takeover:

- $\alpha$ incumbent
- $\beta$ entrant
- passing the output of $\alpha$ by $\varphi_1$ is the same as passing the input to $\beta$ by $\varphi_0$
**Contract dynamics**

**forward:** probability that $\alpha$ is overtaken by $\beta$
(and not by another entrant)

\[
\tilde{A}_\alpha^\triangleright = \frac{\tilde{A}_{\alpha\beta}}{\tilde{A}_\alpha^\bullet}
\]

**backward:** probability $\beta$ overtakes $\alpha$
(and not another incumbent)

\[
\tilde{A}_\alpha^\triangleleft = \frac{\tilde{A}_{\alpha\beta}}{\tilde{A}_\bullet^\beta}
\]
**Contract ranking**

**push rank:** weight of $\alpha$’s contract obligations

$$\tilde{r}_\alpha = \sum_{\beta \in \tilde{N}} \tilde{A}_{\alpha \beta} \tilde{r}_\beta$$

**pull rank:** weight of $\beta$’s contract entitlements

$$\tilde{r}_\beta = \sum_{\alpha \in \tilde{N}} \tilde{r}_\alpha \tilde{A}_{\alpha \beta}$$
Contract flow

from $k$ to $\ell$ is the total entitlement of all contracts with $k$ as the seller and $\ell$ as the buyer

\[ R_{k\ell} = \sum_{k \rightarrow \ell} \bar{r}_\beta \]
**Proposition.** If a contract network $A$ supports fixed pull-competition, the right marginal of the contract flow distribution is equal to the pull rank:

$$
\sum_{k \in N} R_{k\ell} = r^p_\ell
$$

If it supports fixed push-competition, then the left marginal of the contract flow distribution equals the push rank:

$$
\sum_{\ell \in N} R_{k\ell} = r^a_k
$$
Recall that

\[ r^\triangledown_\ell = \text{Prob}(\varrho(\alpha) = \ell \mid \alpha \in E) \]
\[ = \text{Prob}(\bullet \xrightarrow{\alpha} \ell \mid \alpha \in E) \]
\[ r^\rhd_i = \text{Prob}(\delta(\alpha) = i \mid \alpha \in E) \]
\[ = \text{Prob}(i \xrightarrow{\alpha} \bullet \mid \alpha \in E) \]

Proposition tells that contract flow is the joint distribution of \( \delta \) and \( \varrho \)

\[ R_{i\ell} = \text{Prob}(\delta(\alpha) = i \land \varrho(\alpha) = \ell \mid \alpha \in E) \]
\[ = \text{Prob}(i \xrightarrow{\alpha} \ell \mid \alpha \in E) \]
Q: How much computation does the network $A$ perform?

- How much non-local information processing is there?

- How much do $\delta$ and $\varrho$ depend on each other?

A: Mutual information of the inputs and the outputs:

\[
I(r^\triangleleft ; r^\triangleright) = D(R || r^\triangleright)
\]

\[
= N \sum_{j=1}^{N} \sum_{k=1}^{N} R_{jk} \log \frac{R_{jk}}{r^\triangleleft_j r^\triangleright_k}
\]

- $I(r^\triangleleft ; r^\triangleright) = 0 \iff R_{jk} = r^\triangleleft_j r^\triangleright_k$

- $I(r^\triangleleft ; r^\triangleright) = H(r) \iff r = r^\triangleleft = r^\triangleright$
Contract flow bias

is the difference of contract flow and expected flow

\[ \gamma_{jk} = R_{jk} - r^\diamond_{jk} \]

it does detect the nodes with unexpected flows
Cohesion of $U \subseteq N$ is the average bias

$$\theta(U) = \frac{\sum_{j,k \in U} \gamma_{jk}}{2|U|}$$

Community is $U \subseteq N$ such that $\theta(U) \geq 0$.

Clique is a community $V$ such that every $U \subseteq V$ is also a community.

Cluster is a clique $W$ such that for every clique $V$ holds

$$V \cap W \neq \emptyset \implies V \subseteq W$$
**Tightness** of $U \subseteq N$

$$\tau(U) = \bigwedge_{X \subseteq U} \theta(X)$$

**Lemma.** $U$ is clique $\iff \tau(U) \geq 0$
Finding clique

Find $V$ such that $\tau(V) \geq \varepsilon$

- **step 1**: search for $j, k \in N$ s.t.
  $\Upsilon_{jk} \land \Upsilon_{kj} \geq \varepsilon$;
  - if there is such a pair, then set $V_1 = \{j, k\}$,
  - else set $V = \emptyset$ and halt.

- **step $i + 1$**: search for $\ell \in N \setminus V_i$ s.t.
  $\Upsilon_{\ell v} \land \Upsilon_{v\ell} \geq \varepsilon$ for every $v \in V_i$;
  - if there is such a node, then set $V_{i+1} = V_i \cup \{\ell\}$,
  - else set $V = V_i$ and halt.
Finding clusters

Search for \( (\equiv) \subseteq N \times N \) maximizing

\[
\Theta(\equiv) = \sum_{j \equiv k} \Upsilon_{jk} - \sum_{j \not\equiv k} \Upsilon_{jk}
\]
Current work

- clustering algorithm (experimentation!).

- tune it up for practical use (empiric data!).
Outline

1. Introduction

2. Survey of link surfing and search

3. Path navigation

4. Communities, cliques and clusters

5. Rank distribution and optimal coding

6. Future work
Theorem. Suppose that a network $A$ evolves in such a way that

- at each step a single node is added, i.e. $A_N$ becomes $A_{N+1}$,

- the rank $r_N(n)$ of a node $n \in N$ changes in $N + 1$-st generation to $r_{N+1}(n)$ according to

$$\text{Prob}\left(r_{N+1}(n) \geq r_N(n) + \frac{1}{N}\right) \sim r_N(n)$$

Then the rank distribution in the resulting network must be scaling, i.e. it obeys the power law

$$v(k) = Ck^{-b}$$
Rank information and storage

Rényi entropy of order $a$ is

$$H^a(v) = \frac{1}{a} \cdot \log \sum_{k=1}^{N} Ck^{a-b}$$

$$\approx \frac{\log C}{1-a} + \frac{1}{1-a} \log \zeta(b-a), \text{ for } b-a > 1$$

Use Blumer-McEliece codes for storage