PageRank Algorithm

Sehyun Kwon

Mathematical Data Science
Hanyang University ERICA

Score redistribution and random walks

Each page divides its score equally among its outgoing links:

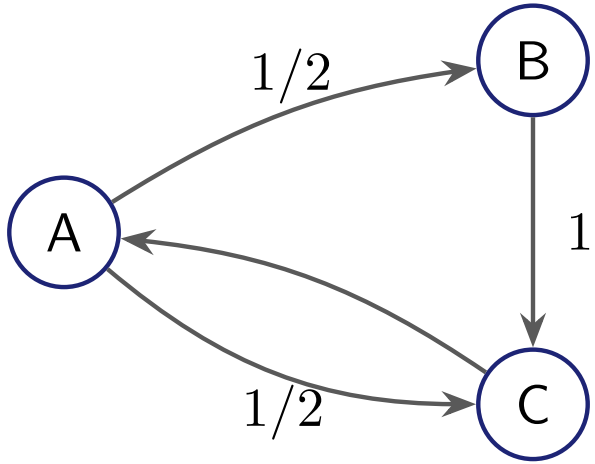
$$r_{t+1}(v) = \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}.$$

Now imagine a user who chooses one outgoing link uniformly at each step.

The probability of being at each page follows **the same update rule**.

A random walk on a directed graph

A **random walk** moves from vertex to vertex by randomly choosing the next step. Here, choose one outgoing link uniformly at each step.

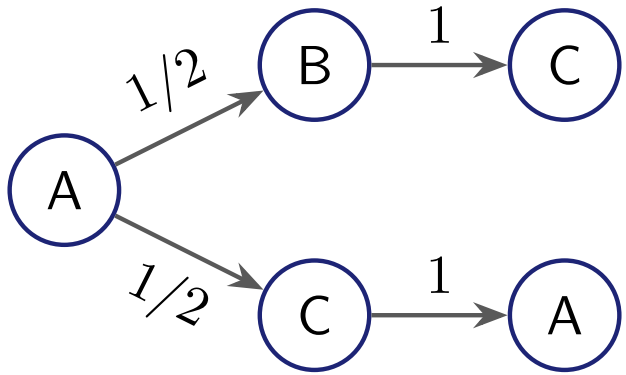


From A, choose B or C with probability $1/2$ each. From B, go to C. From C, go to A.

One possible walk is $A, B, C, A, C, A, B, \ldots$. The walker may visit the same page multiple times. We are interested in how often it visits each page.

Where is the walker after two steps?

Problem. Start at A. Find the location probabilities after one step and after two steps.



Solution. Write X_t for the page occupied after t steps.

	$P(X_t = A)$	$P(X_t = B)$	$P(X_t = C)$
$t = 0$	1	0	0
$t = 1$	0	$1/2$	$1/2$
$t = 2$	$1/2$	0	$1/2$

X_t records a page label.

Location probabilities follow the basic rule

Set $r_t(u) = P(X_t = u)$. Condition on the walker's current page:

$$P(X_{t+1} = v) = \sum_{u \in V} P(X_t = u) P(X_{t+1} = v \mid X_t = u).$$

If $u \to v$, the second factor is $1/d^+(u)$. Otherwise it is zero.

$$\boxed{r_{t+1}(v) = \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}.}$$

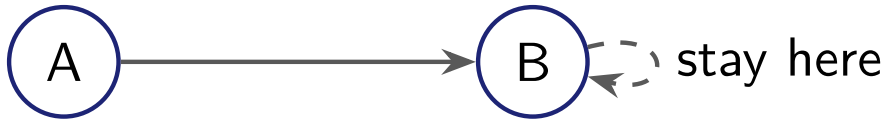
For our graph, for example,

$$P(X_{t+1} = C) = \tfrac{1}{2} P(X_t = A) + P(X_t = B).$$

With a uniform starting page, these probabilities are exactly the score tables from the previous lecture. Successive locations generally depend on each other.

A page with no outgoing links

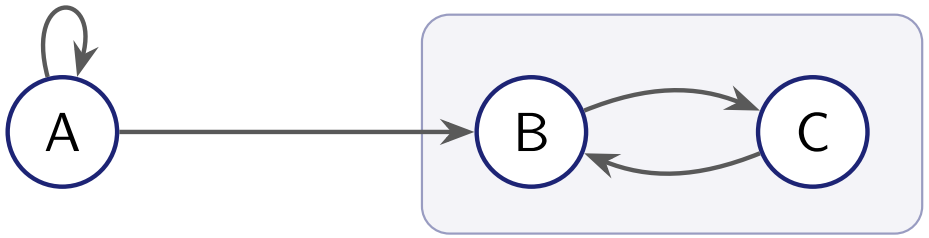
A page with out-degree zero is called a **dangling node**. Its score has nowhere to go under the basic rule.



Our convention: the walker stays at a dangling node. For computation, give it a self-loop $B \to B$.

Score stays there too. This preserves total score but leaves the walker trapped.

A closed group can trap score



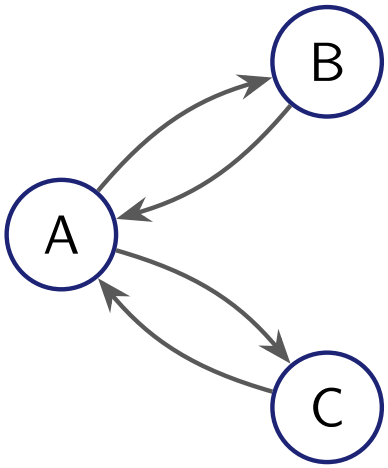
Once the walker enters $\{B, C\}$,
it cannot leave.

Start with $1/3$ at each node. Node A sends half its score to itself and half to B; nodes B and C only send to each other.

$$r_t(A) = \frac{1}{3}\left(\frac{1}{2}\right)^t \longrightarrow 0, \qquad r_t(B) + r_t(C) \longrightarrow 1.$$

Eventually almost all score lies in $\{B, C\}$. A region can trap score even when no node has out-degree zero.

Scores need not settle



t	A	B	C
0	$1/3$	$1/3$	$1/3$
1	$2/3$	$1/6$	$1/6$
2	$1/3$	$1/3$	$1/3$
3	$2/3$	$1/6$	$1/6$

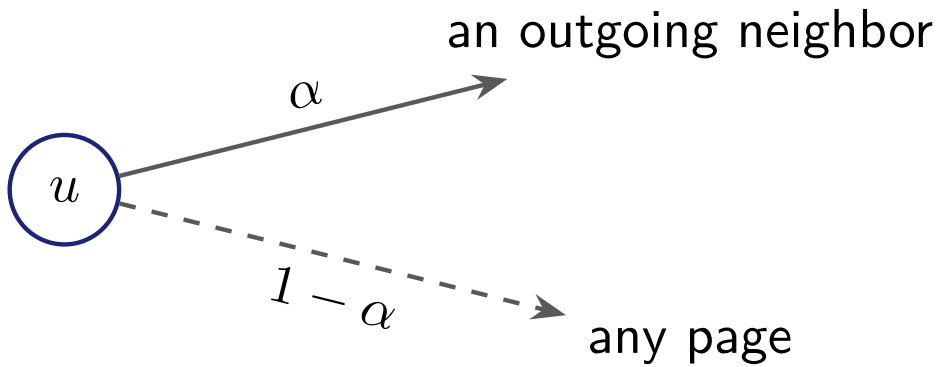
Score alternates between the center and the two outer nodes. Even strong connectivity does not guarantee that the basic iterates converge.

There is an equilibrium $(1/2, 1/4, 1/4)$, but these iterates do not approach it.

A random walk with teleportation

Choose $0 < \alpha < 1$. At each step, make a fresh random choice:

- With probability α, follow one outgoing link uniformly. At a dangling node, stay there on this branch.
- With probability $1 - \alpha$, jump to a uniformly chosen page. The destination may be the current page.



This jump is **teleportation**. It lets the walker leave a closed group or a dangling node without an outgoing link.

Deriving the update: teleportation

Let $r_t(u) = P(X_t = u)$. At the next step, let J mean teleportation and L mean following a link. By the **law of total probability**,

$$P(X_{t+1} = v) = (1-\alpha)P(X_{t+1} = v \mid J) + \alpha P(X_{t+1} = v \mid L).$$

A jump can reach v from **any starting page** u. The choice to jump is independent of the current page, so

$$(1-\alpha)P(X_{t+1} = v \mid J) = \sum_{u \in V} \underbrace{r_t(u)}_{\text{start at } u} \underbrace{(1-\alpha)}_{\text{jump}} \underbrace{\frac{1}{n}}_{\text{choose } v}$$

$$= \frac{1-\alpha}{n} \underbrace{\sum_{u \in V} r_t(u)}_{=1} = \frac{1-\alpha}{n}.$$

Deriving the update: following links

For a link move to reach v, the starting page u must have a link to v. Sum over these starting pages:

$$\alpha P(X_{t+1} = v \mid L) = \sum_{u:\, u \to v} \underbrace{r_t(u)}_{\text{start at } u} \quad \underbrace{\alpha}_{\text{follow a link}} \quad \underbrace{\frac{1}{d^+(u)}}_{\text{choose } v}.$$

Here the graph includes the self-loops added at dangling nodes.

Adding the teleportation and link contributions gives

$$r_{t+1}(v) = \underbrace{\frac{1-\alpha}{n}}_{\text{arrive by teleportation}} + \underbrace{\alpha \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}}_{\text{arrive by following a link}}.$$

The weights $1 - \alpha$ and α are the probabilities of choosing the two movement rules.

Scaled PageRank

Choose a number α with $0 < \alpha < 1$. Modify each round in two steps:

1. Redistribute score along links, then multiply it by α.
2. Spread the remaining amount $1 - \alpha$ equally over all n pages.

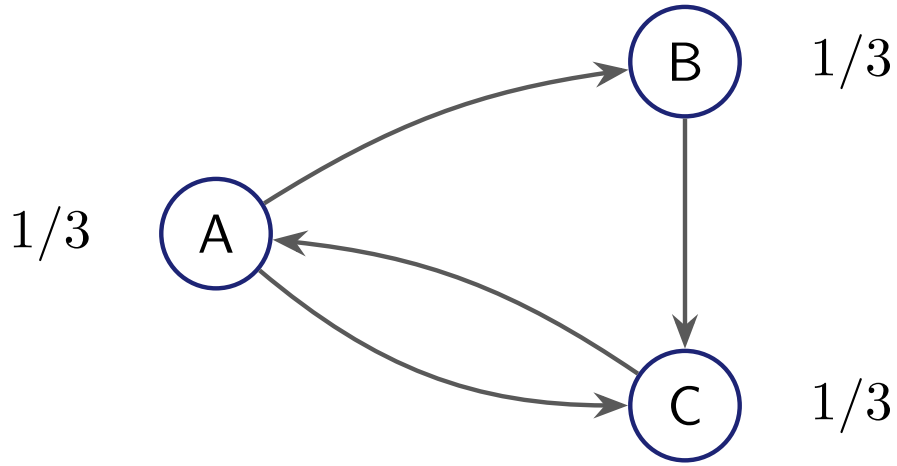
$$\boxed{r_{t+1}(v) = \frac{1-\alpha}{n} + \alpha \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}.}$$

Use the graph after adding a self-loop to each dangling node.

We use $\alpha = 0.85$ in our numerical example. Every page receives at least $(1-\alpha)/n$ each round.

One scaled update by hand

Return to our original three-page graph and start uniformly.



The basic update gives $(1/3, 1/6, 1/2)$. With $\alpha = 0.85$, each page also receives $0.15/3 = 0.05$.

$$r_1(A) = 0.05 + 0.85(1/3) \approx 0.3333,$$
$$r_1(B) = 0.05 + 0.85(1/6) \approx 0.1917,$$
$$r_1(C) = 0.05 + 0.85(1/2) = 0.4750.$$

Why the scores still sum to one

Suppose $\sum_{u \in V} r_t(u) = 1$.

Teleportation. Each of the n pages receives $(1-\alpha)/n$:

$$\sum_{v \in V} \frac{1-\alpha}{n} = n \frac{1-\alpha}{n} = 1 - \alpha.$$

Following links. Page u divides $\alpha r_t(u)$ among its $d^+(u)$ outgoing links, so its total contribution is

$$d^+(u) \frac{\alpha r_t(u)}{d^+(u)} = \alpha r_t(u).$$

Every contribution is received by a page. Summing over all senders,

$$\text{total link contribution} = \sum_{u \in V} \alpha r_t(u) = \alpha.$$

Hence $\boxed{\sum_{v \in V} r_{t+1}(v) = (1-\alpha) + \alpha = 1.}$

Starting with $r_0(v) = 1/n$ preserves this sum in every round. Self-loops at dangling nodes ensure that no score is lost.

Why teleportation helps

The walker can escape a closed group. Even if no link leads outside the group, teleportation can take the walker to a page outside it.

The scores converge to the same equilibrium. For a finite graph with uniform teleportation and $0 < \alpha < 1$, every initial score distribution converges to the same limit. There is exactly one distribution with total score 1 that remains unchanged by an update.

This uses our self-loop rule for dangling nodes. We will prove this result next.

α **controls how the walker moves.** A larger α means following links more often. A smaller α means teleporting more often.

Two pages: the update and numerical example

For $A \to B$ and $B \to A$, write $r_t = (r_t(A), r_t(B))$.

$$r_{t+1}(A) = \frac{1-\alpha}{2} + \alpha r_t(B), \qquad r_{t+1}(B) = \frac{1-\alpha}{2} + \alpha r_t(A).$$

Without teleportation, $(1,0)$ and $(0,1)$ alternate forever. With $\alpha = 1/2$ and $r_0 = (1,0)$:

$$D_t = |r_t(A) - r_{t-1}(A)| + |r_t(B) - r_{t-1}(B)|, \quad t \geq 1.$$

t	$r_t(A)$	$r_t(B)$	D_t
0	1	0	—
1	0.25	0.75	1.5
2	0.625	0.375	0.75
3	0.4375	0.5625	0.375
4	0.53125	0.46875	0.1875

For example, $D_2 = |0.625 - 0.25| + |0.375 - 0.75| = 0.75$.

Two pages: shrinking changes and convergence

Subtract consecutive updates. The teleportation terms cancel:

$$\begin{aligned} r_{t+1}(A) - r_t(A) &= \left(\frac{1-\alpha}{2} + \alpha r_t(B)\right) - \left(\frac{1-\alpha}{2} + \alpha r_{t-1}(B)\right) \\ &= \alpha(r_t(B) - r_{t-1}(B)), \\ r_{t+1}(B) - r_t(B) &= \alpha(r_t(A) - r_{t-1}(A)). \end{aligned}$$

Thus

$$\begin{aligned} D_{t+1} &= \alpha|r_t(B) - r_{t-1}(B)| + \alpha|r_t(A) - r_{t-1}(A)| \\ &= \alpha D_t, \qquad \text{so } D_t = \alpha^{t-1} D_1. \end{aligned}$$

To find the limit, use $r_t(B) = 1 - r_t(A)$:

$$\begin{aligned} r_{t+1}(A) - \tfrac{1}{2} &= \tfrac{1-\alpha}{2} + \alpha(1 - r_t(A)) - \tfrac{1}{2} \\ &= \tfrac{\alpha}{2} - \alpha r_t(A) = -\alpha(r_t(A) - \tfrac{1}{2}), \\ r_t(A) - \tfrac{1}{2} &= (-\alpha)^t (r_0(A) - \tfrac{1}{2}) \longrightarrow 0. \end{aligned}$$

Hence $r_t(A), r_t(B) \to 1/2$ for any initial distribution. The limit is fixed: $(1-\alpha)/2 + \alpha(1/2) = 1/2$.

General graphs: the change contracts

Let V have n pages, $0 < \alpha < 1$, and $\sum_v r_0(v) = 1$ with $r_0(v) \geq 0$. Add self-loops at dangling nodes.

$$r_{t+1}(v) = \frac{1-\alpha}{n} + \alpha \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}, \qquad D_t = \sum_v |r_t(v) - r_{t-1}(v)|.$$

Subtract updates, apply the triangle inequality, and count by sender:

$$r_{t+1}(v) - r_t(v) = \alpha \sum_{u:\, u \to v} \frac{r_t(u) - r_{t-1}(u)}{d^+(u)}.$$

$$\begin{aligned} D_{t+1} &= \alpha \sum_v \left| \sum_{u:\, u \to v} \frac{r_t(u) - r_{t-1}(u)}{d^+(u)} \right| \\ &\leq \alpha \sum_v \sum_{u:\, u \to v} \frac{|r_t(u) - r_{t-1}(u)|}{d^+(u)} \\ &= \alpha \sum_u d^+(u) \frac{|r_t(u) - r_{t-1}(u)|}{d^+(u)} = \alpha D_t. \end{aligned}$$

Each sender's term occurs $d^+(u)$ times, once per outgoing link. Therefore $\boxed{D_t \leq \alpha^{t-1} D_1}$ for $t \geq 1$.

Why the scores converge

The fact that $D_t \to 0$ alone would not be enough. Here, the **sum of all remaining changes** becomes small.

For any $m \geq 1$, telescoping and the triangle inequality give

$$\begin{aligned}
\sum_v |r_{t+m}(v) - r_t(v)| &= \sum_v \left| \sum_{j=1}^{m} (r_{t+j}(v) - r_{t+j-1}(v)) \right| \\
&\leq \sum_{j=1}^{m} D_{t+j} \\
&\leq D_1 \sum_{j=1}^{m} \alpha^{t+j-1} \\
&\leq \frac{D_1 \alpha^t}{1-\alpha} \longrightarrow 0.
\end{aligned}$$

This bound holds no matter how many more updates we perform. The total possible change after round t approaches zero. Thus each page's score converges to a limit, denoted $r_*(v)$.

The limit is an equilibrium

Take limits in the update equation:

$$\begin{aligned} r_*(v) &= \lim_{t\to\infty} r_{t+1}(v) \\ &= \frac{1-\alpha}{n} + \alpha \sum_{u:\, u\to v} \frac{\lim_{t\to\infty} r_t(u)}{d^+(u)} \\ &= \frac{1-\alpha}{n} + \alpha \sum_{u:\, u\to v} \frac{r_*(u)}{d^+(u)}. \end{aligned}$$

The sum is finite, so limits may be taken term by term. This equation says that updating r_* leaves it unchanged.

Also,

$$r_*(v) \geq 0, \qquad \sum_v r_*(v) = \lim_{t\to\infty} \sum_v r_t(v) = 1.$$

Thus the limit is an equilibrium score distribution.

Why the equilibrium is unique

For any distributions p, q, let p', q' be their updated scores. Define $d(p,q) = \sum_v |p(v) - q(v)|$. As before,

$$\begin{aligned} d(p',q') &= \alpha \sum_v \left| \sum_{u:\, u\to v} \frac{p(u)-q(u)}{d^+(u)} \right| \\ &\le \alpha \sum_v \sum_{u:\, u\to v} \frac{|p(u)-q(u)|}{d^+(u)} \\ &= \alpha \sum_u d^+(u) \frac{|p(u)-q(u)|}{d^+(u)} = \alpha d(p,q). \end{aligned}$$

If p, q are both equilibria, then $p' = p,\ q' = q$, so

$$d(p,q) \le \alpha d(p,q) \quad \Longrightarrow \quad (1-\alpha) d(p,q) \le 0.$$

Since $1 - \alpha > 0$ and $d(p,q) \ge 0$,

$$d(p,q) = \sum_v |p(v) - q(v)| = 0.$$

Every term is nonnegative, hence $p(v) = q(v)$ for every v.

Every initial distribution therefore converges to the unique equilibrium.

Reading a PageRank score

At equilibrium, $r(v)$ is the probability of finding the random surfer at page v when their location has the equilibrium distribution.

It also describes the long-run fraction of visits to v in this model. It is not the probability that the page contains a correct answer.

To connect this with expectation, define

$$I_t = \begin{cases} 1 & X_t = v, \\ 0 & X_t \neq v. \end{cases}$$

If the walk starts at equilibrium, then for m recorded steps,

$$\mathbb{E}\left[\sum_{t=1}^{m} I_t\right] = m\, r(v).$$

Linearity of expectation applies even though successive visits are dependent.

The effect of α

α is the probability of following a link at each step. It controls how strongly the scores depend on the links.

α	$r(A)$	$r(B)$	$r(C)$
0	0.3333	0.3333	0.3333
0.50	0.3590	0.2564	0.3846
0.85	0.3878	0.2148	0.3974
0.95	0.3960	0.2048	0.3993

These are equilibrium scores for our original three-page graph. At $\alpha = 0$, every step is a uniform jump.

The value 0.85 is a common choice, not a mathematical requirement. In a project, compare several choices to see whether the ranking is stable.

When should the computation stop?

Compare consecutive rounds using their total absolute change:

$$\Delta_t = \sum_{v \in V} |r_{t+1}(v) - r_t(v)|.$$

1. Start with equal scores.
2. Compute a complete new round.
3. Stop when $\Delta_t < \varepsilon$, for a chosen tolerance.

Also set a maximum number of rounds and report failure if it is reached. Small change is a numerical stopping criterion, not exact equality.

Check that scores remain nonnegative and sum to approximately one.

Representing the graph in Python

```
links = {
    "A": ["B", "C"],
    "B": ["C"],
    "C": ["A"],
}

nodes = list(links)
n = len(nodes)
out = {u: links[u] or [u] for u in nodes}
r = {u: 1 / n for u in nodes}
```

Every page must be a dictionary key, even if its list is empty. Outgoing lists contain distinct existing pages.

The expression `links[u] or [u]` uses a self-loop only when the original list is empty.

Computing a new round

```
alpha = 0.85
new = {v: (1 - alpha) / n for v in nodes}

for u in nodes:
    share = alpha * r[u] / len(out[u])
    for v in out[u]:
        new[v] += share

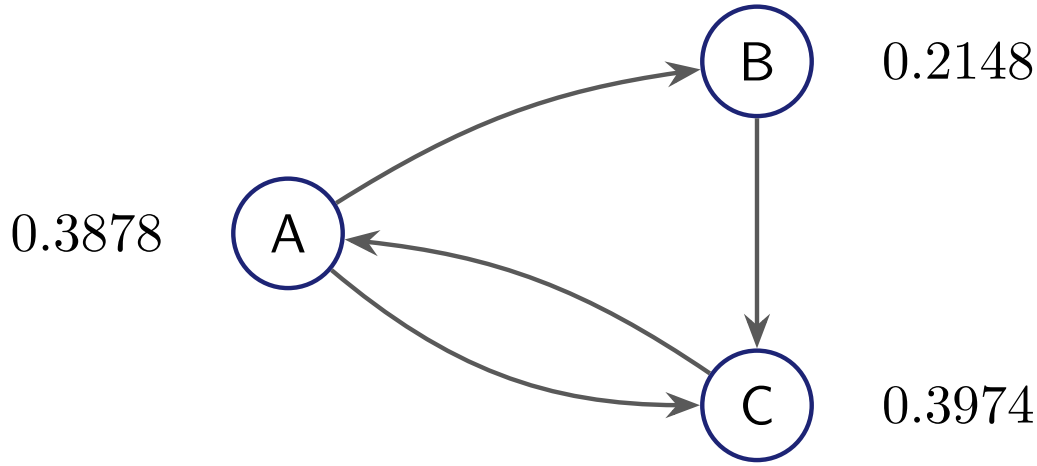
change = sum(abs(new[v] - r[v]) for v in nodes)
r = new
```

First give every page its teleportation share. Then send each page's link share along its outgoing edges.

Repeat this block until the change is small enough. The old scores are not overwritten during distribution.

The computed result

For our three-page graph with $\alpha = 0.85$, iteration gives



$$C > A > B.$$

For example, the incoming contributions to C satisfy

$$r(C) = 0.05 + 0.85 \left(\frac{r(A)}{2} + r(B) \right) \approx 0.3974.$$

The scores sum to one before rounding. The order differs slightly from the basic equilibrium, where A and C were tied.

Simulating visits

We can estimate the scores by simulating the random walk with teleportation. Here, `nodes` and `out` are the same as in the iteration code.

```
import random
rng = random.Random(0)
current = "A"
counts = {v: 0 for v in nodes}
m = 100000
for _ in range(m):
    if rng.random() < 0.85:
        current = rng.choice(out[current])
    else:
        current = rng.choice(nodes)
    counts[current] += 1
frequency = {v: counts[v] / m for v in nodes}
```

Record the location *after* each move. A stay at the same page also counts. The fixed random seed makes this example reproducible.

Visits and computed scores

On our original graph, start the simulation at A and use $\alpha = 0.85$.

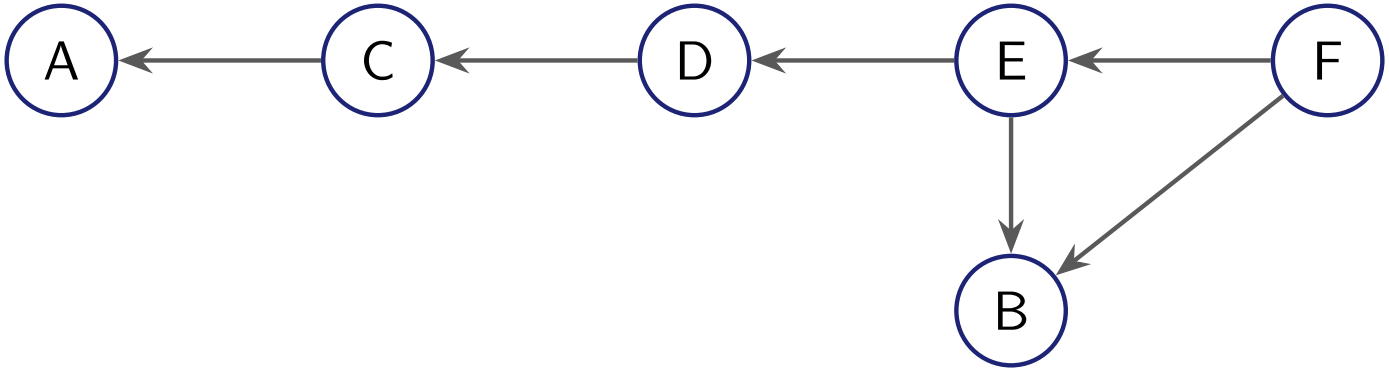
	A	B	C
100 recorded visits	0.3700	0.2200	0.4100
1,000 recorded visits	0.3890	0.2080	0.4030
100,000 recorded visits	0.3882	0.2147	0.3972
PageRank iteration	0.3878	0.2148	0.3974

Each entry in a simulation row is a visit count divided by the number of visits. The rows use prefixes of the same simulated walk (seed 0).

In the long run, visit fractions approach the PageRank scores. A finite run has random error, and more steps need not improve every estimate each time.

Application: choosing papers to read

Problem. In a collection of six papers, which papers receive the most attention through citations? Compare citation counts with PageRank.



A node is a paper. An edge $u \to v$ means that u cites v. These are fictional papers and all displayed citations are equally weighted.

Paper B has two incoming citations. Each of A, C, D, E has one, and F has none. Citation count ranks B first.

The citation graph: model and result

```
links = {
    "A": [],
    "B": [],
    "C": ["A"],
    "D": ["C"],
    "E": ["B", "D"],
    "F": ["B", "E"],
}
```

Paper	In-degree	PageRank
A	1	0.5017
B	2	0.3384
C	1	0.0591
D	1	0.0401
E	1	0.0356
F	0	0.0250

Solution. Apply the same algorithm with $\alpha = 0.85$. Add self-loops at A and B for the link-following step.

The empty lists mean no outgoing citations *within this collection*. They do not imply that the original papers have no references.

PageRank ranks A above B, despite its smaller citation count.

Why does paper A rank first?

The citation chain $F \to E \to D \to C \to A$ carries attention toward A. With our dangling convention, a walker at A stays until teleporting.

For six papers, the teleportation share is $0.15/6 = 0.025$. At equilibrium,

$$r(A) = 0.025 + 0.85\big(r(C) + r(A)\big).$$

Thus, with $r(C) \approx 0.05912$,

$$r(A) = \frac{0.025 + 0.85r(C)}{0.15} \approx 0.5017.$$

This is attention under our model, not a guarantee of paper quality. The collection boundary and the dangling rule strongly affect this example. Different dangling rules can give different scores.

Recommendations for different users

Problem. A football fan and a gaming fan use the same social network. Which accounts should we recommend to each person?

With uniform teleportation, PageRank gives one ranking of all accounts. Every user would receive the same ranking.

For personal recommendations, we want scores that depend on **whose interests we are considering**.

Personalized PageRank changes where the random walker returns when teleporting.

Teleportation back to a user

To compute scores for user A, use the following rule at every step:

- With probability α, follow an outgoing link uniformly.
- With probability $1 - \alpha$, return to A.

At a dangling node, keep our earlier convention: stay there on the link-following branch.

With $\alpha = 0.85$, the walker follows links with probability 0.85 and returns to A with probability 0.15.

We return to A repeatedly throughout the walk. Changing only the starting location would leave the usual PageRank equilibrium unchanged.

The personalized update

Let $q(v)$ be the probability of teleporting to node v. The update becomes

$$r_{t+1}(v) = (1-\alpha)q(v) + \alpha \sum_{u:\, u \to v} \frac{r_t(u)}{d^+(u)}.$$

As before, add self-loops at dangling nodes for the link-following step.

For uniform teleportation, $q(v) = 1/n$. For user A, choose

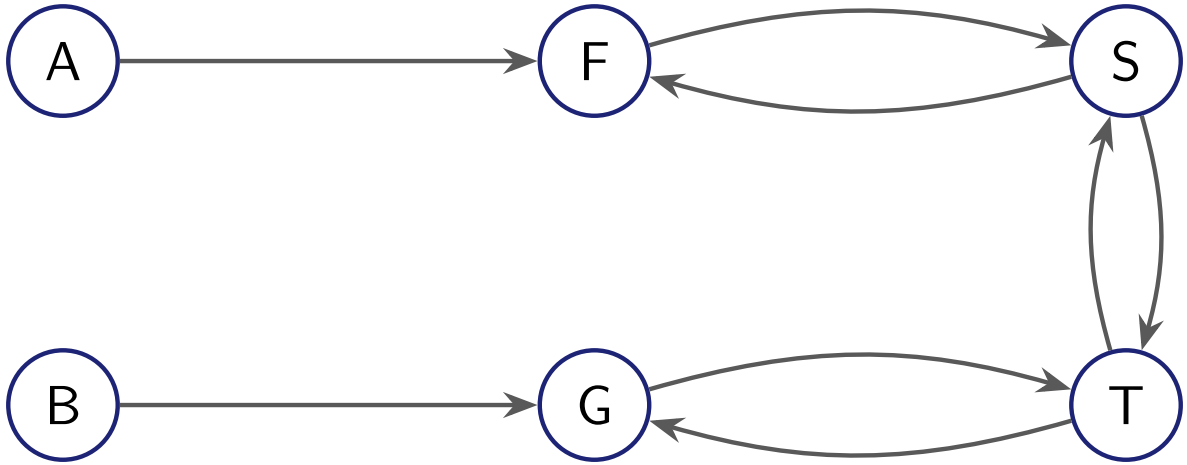
$$q(v) = \begin{cases} 1 & v = A, \\ 0 & v \neq A. \end{cases}$$

Each round gives the extra $1-\alpha$ to A. The link shares and the stopping rule stay the same.

In the code, initialize `new[v]` with `(1-alpha)*q[v]`.

A small follow network

Each node is an account. An arrow $u \to v$ means that u follows v.



A: football fan
B: gaming fan
F: football news
G: gaming news
S: football analyst
T: game streamer

A already follows F. B already follows G. Which additional accounts should we recommend?

Same network, different recommendations

Iterate with $\alpha = 0.85$, teleporting to A or to B. The equilibrium scores are approximately:

Account	For user A	For user B
A	0.1500	0.0000
B	0.0000	0.1500
F	0.2569	0.0861
G	0.0861	0.2569
S	0.3044	0.2026
T	0.2026	0.3044

Exclude the user and accounts they already follow. Rank the remaining accounts using their scores.

$$\text{For } A: \quad S > T > G > B, \qquad \text{for } B: \quad T > S > F > A.$$

Recommend S to the football fan and T to the gaming fan.

This fictional example is a simplified recommendation model, not a reproduction of Twitter's full recommendation system. Personalized scores can be zero for unreachable accounts.