

Master Your Own Recommender Systems

Rule-Based Recommendation, Markov Models and PageRank

Institute of Communication and Marketing IKM
CC Communication and Marketing Technologies

Dr. Guang Lu
Lecturer

T direct +41 41 228 22 51
guang.lu@hslu.ch


Lucerne 08.01.2021

What We Are Going to Learn

- Rule-based recommender systems, e.g., Hacker News, Reddit
- Markov models and PageRank, e.g., Google Search



Hacker News Rule-Based Recommendation

 **Hacker News** | new | past | comments | ask | show | jobs | submit

1. ▲ **FCC will require phone carriers to authenticate calls by June 2021** [pdf] (fcc.gov)
734 points by hbcondo714 11 hours ago | hide | 271 comments
2. ▲ **Omni Group Layoffs** (mjtsai.com)
154 points by keehun 5 hours ago | hide | 77 comments
3. ▲ **Simdjson 0.3: Faster JSON parser** (lemire.me)
109 points by ngaut 4 hours ago | hide | 19 comments
4. ▲ **Trolls break into meetings on Zoom** (businessinsider.com)
314 points by pseudolus 10 hours ago | hide | 202 comments
5. ▲ **Vim rendered on a cube for no reason** (github.com)
316 points by ohjeez 9 hours ago | hide | 80 comments
6. ▲ **Honda bucks industry trend by removing touchscreen controls** (autocar.co.uk)
1590 points by trenning 16 hours ago | hide | 632 comments
7. ▲ **A startup is building computer chips using real neurons** (fortune.com)
55 points by gautamsivakumar 3 hours ago | hide | 21 comments
8. ▲ **Planning and Managing Layoffs** (a16z.com)
176 points by todsacerdoti 8 hours ago | hide | 61 comments
9. ▲ **Apple acquires Dark Sky** (darksky.net)
779 points by LittleAthena 14 hours ago | hide | 519 comments
10. ▲ **Writing HTML with Racket and X-Expressions (2019)** (xy2.dev)
27 points by xy2_ 3 hours ago | hide | 15 comments
11. ▲ **The origin of the colors in the first color photographs** (phys.org)
5 points by vo2maxer 1 hour ago | hide | discuss

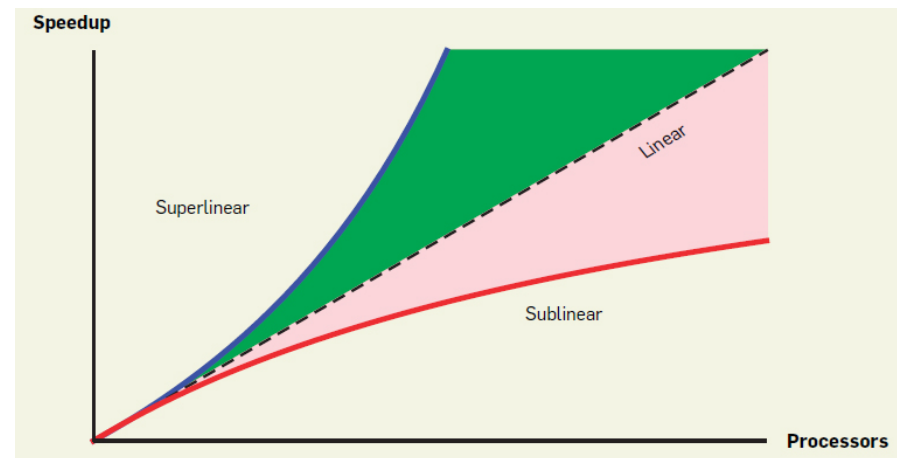
$$\frac{f(\text{popularity})}{g(\text{time})}$$

Hacker News Formula

$$score = \frac{(ups - downs - 1)^{0.8}}{(age + 2)^{gravity}} \times penalty$$

$$gravity = 1.8$$

<https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d>
<http://www.righto.com/2013/11/how-hacker-news-ranking-really-works.html>

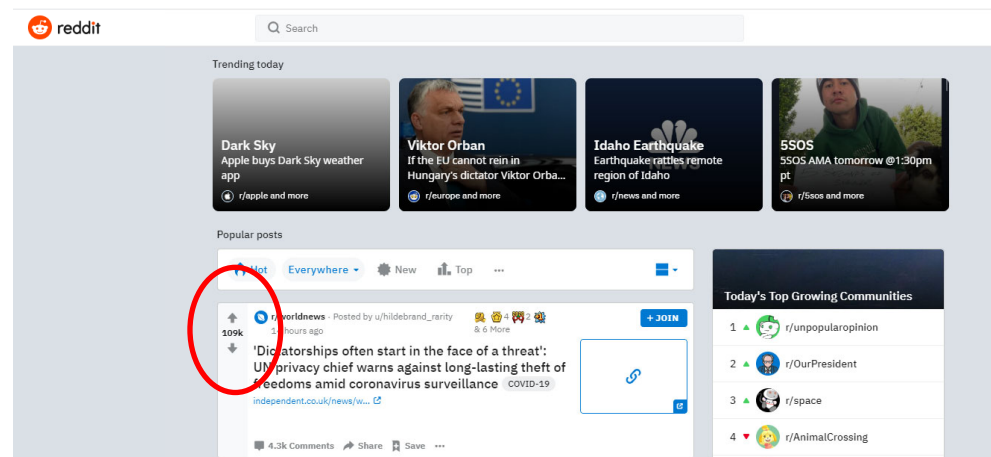


Comparing to Reddit Rule-Based Recommendation

$$score = sign(ups - downs) \times \log\{\max(1, |ups - downs|)\} + \frac{age}{45000}$$

age is in seconds counted from the inception of Reddit (08.12.2005)

<https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9>
<https://redditblog.com/2009/10/15/reddits-new-comment-sorting-system/>



Markov Models

- A generic state at time t can be defined as $x(t)$
- For Markov models, $x(t)$ does not depend on any states that are two or more time steps in the past, but only depends on the immediate last state $x(t-1)$

$$p(x_t | x_{t-1}, x_{t-2}, \dots, x_0) = p(x_t | x_{t-1})$$

<https://hackernoon.com/from-what-is-a-markov-model-to-here-is-how-markov-models-work-1ac5f4629b71>

Markov Chains

Markov chains are mathematical systems that hop from one «state» (a situation or a set of values) to another. Examples of markov chains are:

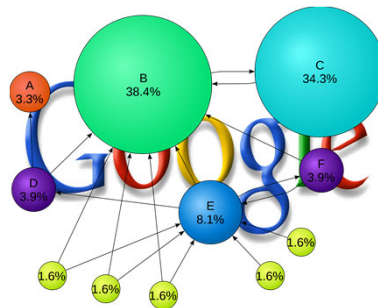
- Weather prediction



- Typing word prediction



- Google PageRank



Markov Chains

Explained Visually

[Twitter](#) [Like 36](#) [Share](#)

By [Victor Powell](#)

with text by [Lewis Lehe](#)

Markov chains, named after [Andrey Markov](#), are mathematical systems that hop from one "state" (a situation or set of values) to another. For example, if you made a Markov chain model of a baby's behavior, you might include "playing," "eating," "sleeping," and "crying" as states, which together with other behaviors could form a "state space": a list of all possible states. In addition, on top of the state space, a Markov chain tells you the probability of hopping, or "transitioning," from one state to any other state—e.g., the chance that a baby currently playing will fall asleep in the next five minutes without crying first.

A simple, two-state Markov chain is shown below.

— speed



<https://setosa.io/ev/markov-chains/>

Transition Probability Matrix

- Transition probability is a probability of going from state i to state j

$$T(i, j) = p(x_t = j | x_{t-1} = i)$$

- «Markov Matrix» or «Transition Matrix»: The probabilities corresponding to the entire M states should sum up to 1

$$\sum_{j=1}^M T(i, j) = \sum_{j=1}^M p(x_t = j | x_{t-1} = i) = 1$$

Calculating Transition Probability

Could you calculate the transition probability for the following two examples?

- Example 1: Weather

sunny, sunny, sunny, rainy, sunny, rainy, rainy, sunny, rainy, rainy, rainy, sunny

- Example 2: Sentence

I think recommender systems are very useful. Therefore I believe this course is perfect for you.

$$p(\text{rainy}|\text{sunny}) = \frac{\text{count}(\text{sunny} \rightarrow \text{rainy})}{\text{count}(\text{sunny})}$$

$$p(\text{think}|I) = \frac{\text{count}(I \rightarrow \text{think})}{\text{count}(I)}$$

$$p(\text{like}|I) = \frac{\text{count}(I \rightarrow \text{like}) + \varepsilon}{\text{count}(I) + \varepsilon\Omega}$$

Ω is vocabulary size

State Distribution I

- State probability distribution at time t is recorded as π_t

$$\pi_t = [p(x_t = \text{sunny}), p(x_t = \text{rainy})]$$

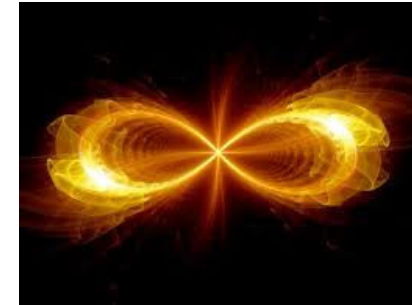
- Future state distributions can be obtained using Bayes rule

$$\begin{aligned} p(x_{t+1} = j) &= \sum_{i=1}^M p(x_{t+1} = j, x_t = i) \\ &= \sum_{i=1}^M p(x_{t+1} = j | x_t = i) p(x_t = i) \\ &= \sum_{i=1}^M T(i, j) \pi_t(i) = \pi_{t+1}(j) \end{aligned}$$

$$\begin{aligned} \pi_{t+1}(j) &= \sum_{i=1}^M T(i, j) \pi_t(i) \\ \pi_{t+1} &= \pi_t T \end{aligned}$$

State Distribution II

- A general form of looking into the future: $\pi_{t+k} = \pi_t T^k$
- What if the time goes to infinity? $\pi_\infty = \lim_{t \rightarrow \infty} \pi_0 T^t$



$$\pi_\infty = \pi_\infty T$$



$$Au = \lambda u$$

«If T is a linear transformation from a vector space V over a field F into itself and v is a nonzero vector in V , then v is an eigenvector of T if $T(v)$ is a scalar multiple of v . This can be written as

$$T(v) = \lambda v,$$

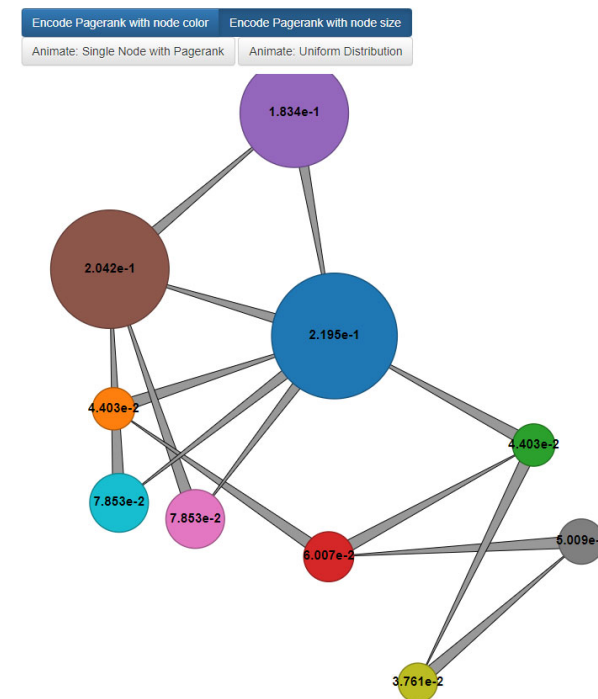
where λ is a scale in F , known as the eigenvalue, characteristic value, or characteristic root associate with v .»

https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

Coming Back to PageRank

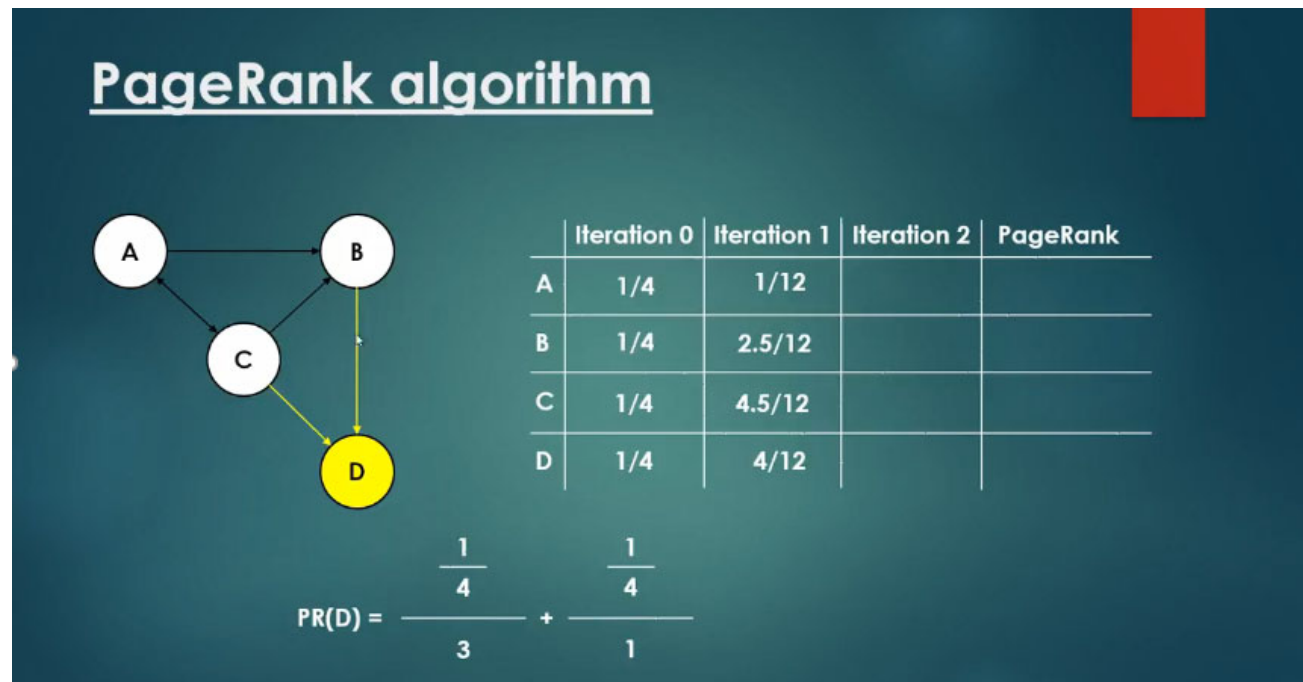
- PageRank is an algorithm used by Google Search to rank web pages in their search engine results
- PageRank is a way of measuring the importance of website pages. According to Google: PageRank works by **counting the number and quality of links to a page** to determine a rough estimate of how important the website is. The underlying assumption is that **more important websites are likely to receive more links from other websites**

<https://en.wikipedia.org/wiki/PageRank>



Pagerank Visualization

A Simple PageRank Example



PageRank and Smoothing

- Two assumptions:
 - ✓ Every page on the Internet is a state in a Markov Model
 - ✓ «Transition Probability» is distributed equally amongst all the links on a page

$$p(x_t = j | x_{t-1} = i) = \begin{cases} \frac{1}{n(i)}, & \text{if } i \text{ links to } j, n(i) \text{ is total number of links on page } i \\ 0, & \text{otherwise} \end{cases}$$

- Smoothing using «Google Matrix» as there are billions of webpages on the Internet:

$$G = 0.85T + 0.15U, U(i, j) = 1/M \quad \forall i, j = 1, \dots, M \quad M \text{ is the total number of states}$$

- Solving the limiting distribution of G leads to a vector of length M , which are the respective PageRanks

$$\pi_{\infty} = \pi_{\infty} G$$

<https://medium.com/@sarthakanand/page-rank-b7072c61dd85>

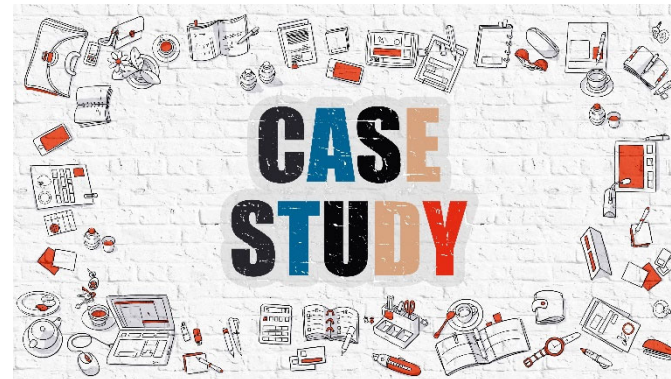
Additional Issues

- Two different concepts in PageRank:
 - ✓ **Limiting distribution:** The state distribution that one would arrive at after transitioning by G an infinite number of times
 - ✓ **Stationary distribution:** A state distribution that does not change after being transitioned by G
- **Perron-Frobenius theorem:** If G is a valid Markov matrix and all its elements are positive, then the stationary distribution and limiting distribution are the same
- How can PageRank be used to solve search problems?
 - ✓ Scenario 1 – Spam the same link many times on the same page: In PageRank, many same links on the page are treated to be equivalent to having one link
 - ✓ Scenario 2 – Create many dummy websites that link back to the page: In PageRank, those dummy websites must be «famous» themselves so that this page can benefit from them
- Search Engine Optimization (SEO)

<https://postcron.com/en/blog/seo-for-blogs-basic-guide/>
<https://postcron.com/en/blog/seo-for-blogs-how-to-improve-pagerank/>

PageRank Case Studies

- The Algorithm behind Google Search: An Implementation with Python
(<https://medium.com/analytics-vidhya/the-algorithm-behind-google-search-an-implementation-with-python-d6418023bbd9>)
- An Introduction to Text Summarization using the TextRank Algorithm with Python Implementation
(<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>)



Summary & What's Next

- We have learned
 - ✓ Rule-based recommendation in Hacker News and Reddit
 - ✓ Markov models and their applications in Google Search
- Recommender system using Bayesian approach



Further Readings

- W.Y. Zhang, S. Zhang, S.S. Guo (2017) A PageRank-based reputation model for personalised manufacturing service recommendation, Enterprise Information Systems, 11:5, 672-693, DOI: 10.1080/17517575.2015.1077998
- M. Richardson, A. Prakash, E. Brill (2006) Beyond PageRank: Machine learning for static ranking, In Proceedings of World Wide Web (WWW) Conference, May 23-26, 2006, Edinburgh, Scotland
- <https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d>
- <http://www.rightho.com/2013/11/how-hacker-news-ranking-really-works.html>
- <https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9>
- <https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a>
- <https://redditblog.com/2009/10/15/reddits-new-comment-sorting-system/>
- <https://ahrefs.com/blog/google-pagerank/>

Reference Video: PageRank Algorithm Explained



https://www.youtube.com/watch?v=P8Kt6Abq_rM