Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Business

# 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

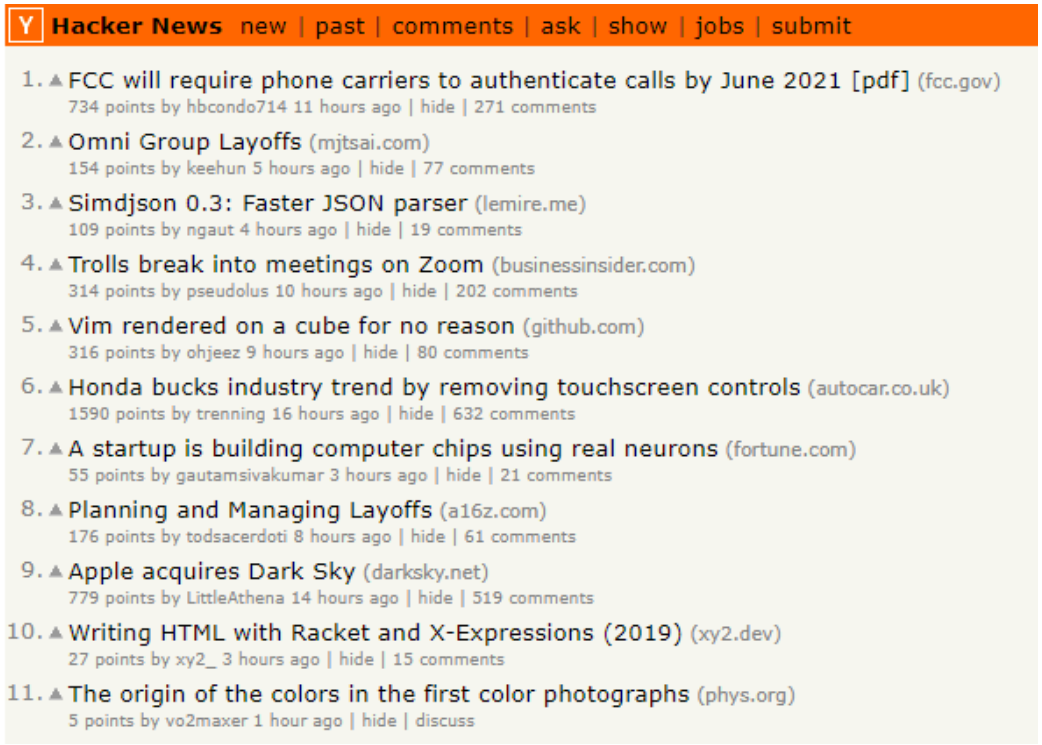
Lucerne 08.01.2021

FH Zentralschweiz

## 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



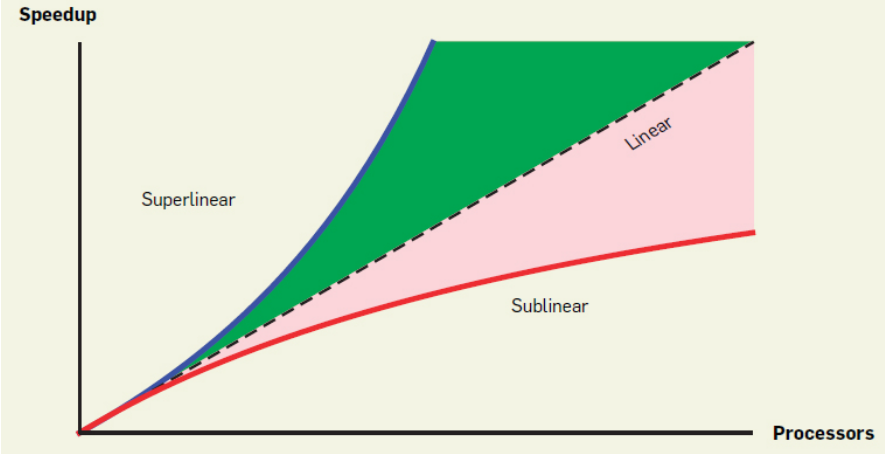

$$\frac{f(popularity)}{g(time)}$$

## 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

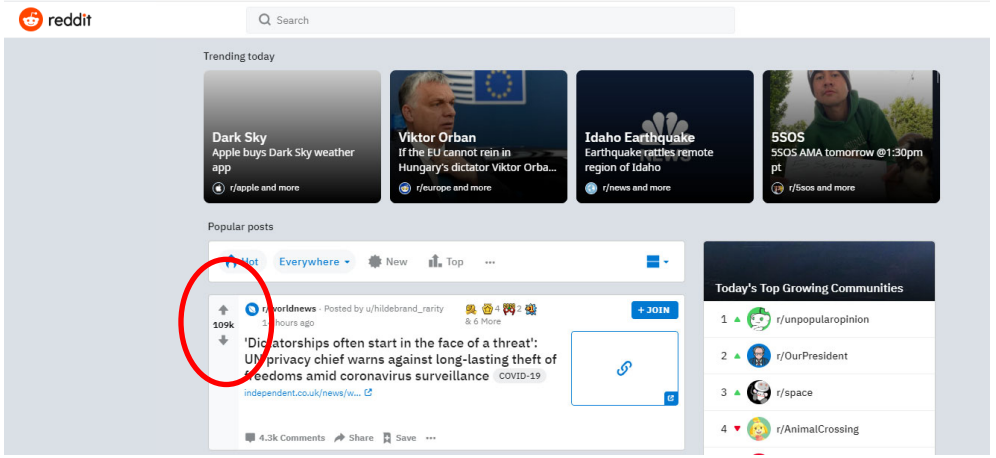


## Comparing to Reddit Rule-Based Recommendation

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

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

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 $x(t-1)$

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

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

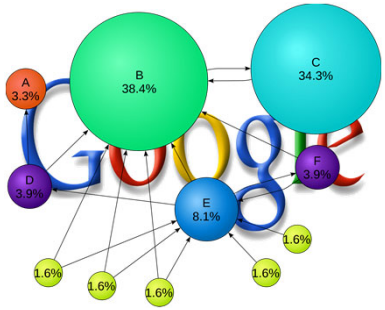
## Markov Chains

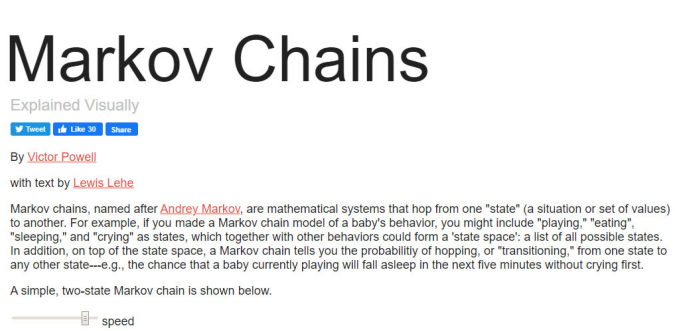
Markov chains are mathematical systems that hop from one «state» (a situation or a set of values) to another. Examples of markov chains are:

- Weather prediction

- Typing word prediction


- Google PageRank




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.

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

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

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

$\Omega$ is vocabulary size

## State Distribution I

- State probability distribution at time $t$ is recorded as $\pi_t$

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

- Future state distributions can be obtained using Bayes rule

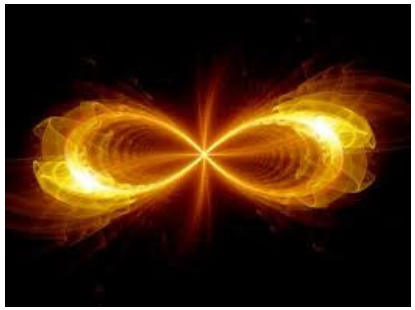
$$p(x_{t+1} = j) = \sum_{i=1}^{M} p(x_{t+1} = j, x_t = i)$$
$$= \sum_{i=1}^{M} p(x_{t+1} = j \,|x_t = i)\, p(x_t = i)$$
$$= \sum_{i=1}^{M} T(i,j)\pi_t(i) = \pi_{t+1}(j)$$

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

$$\pi_{t+1} = \pi_t T$$

## 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? $\pi_\infty = \lim_{t\to\infty} \pi_0 T^t$



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

$$Au = \lambda u$$

«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 $\lambda$ is a scale in $F$, known as the eigenvalue, characteristic value, or characteristic root associate with $v$.»

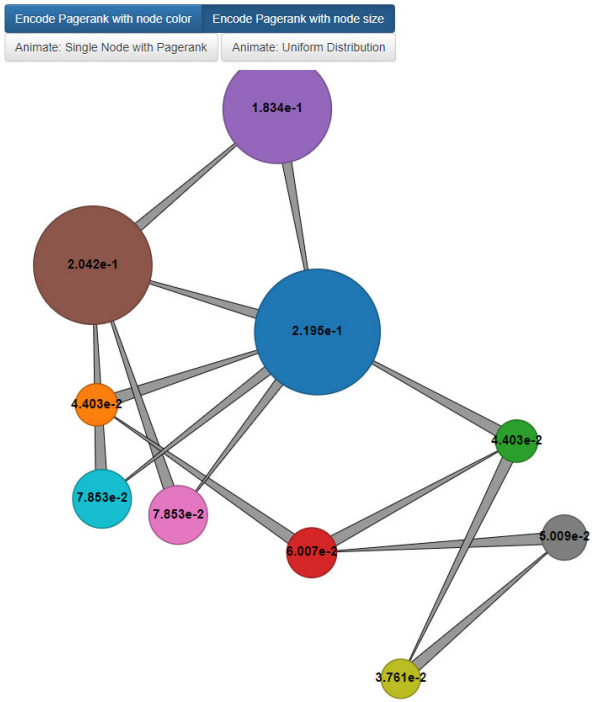
https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

## Coming Back to PageRank

- PageRank is an algorithm used by Google Search to rank web pages in their search engine results

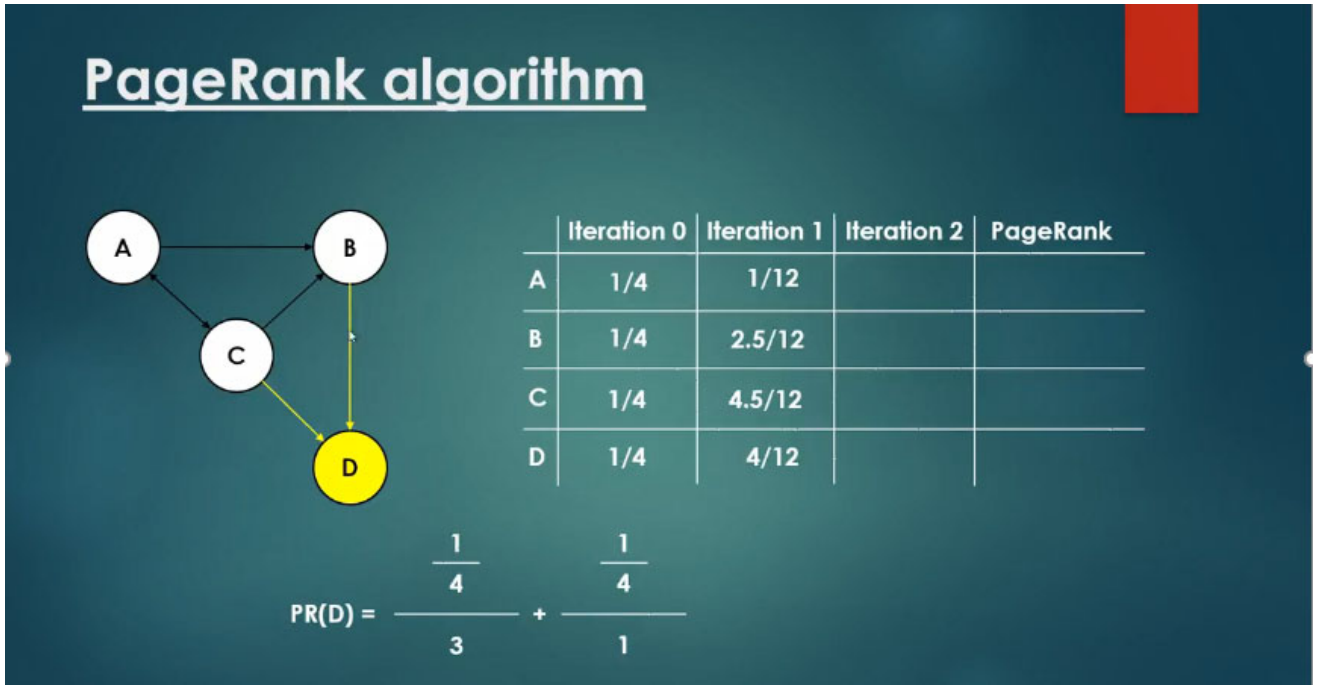
- PageRank is a way of measuring the importance of website pages. According to Google: PageRank works by **counting the number and quality of links to a page** to determine a rough estimate of how important the website is. The underlying assumption is that **more important websites are likely to receive more links from other websites**

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




Pagerank Visualization

## A Simple PageRank Example



## PageRank and Smoothing

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

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

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

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

- Solving the limiting distribution of $G$ leads to a vector of lengh $M$, 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

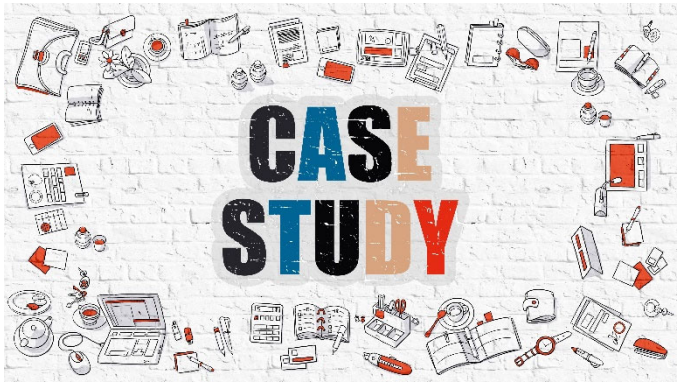
- Search Engine Optimization (SEO)

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.righto.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