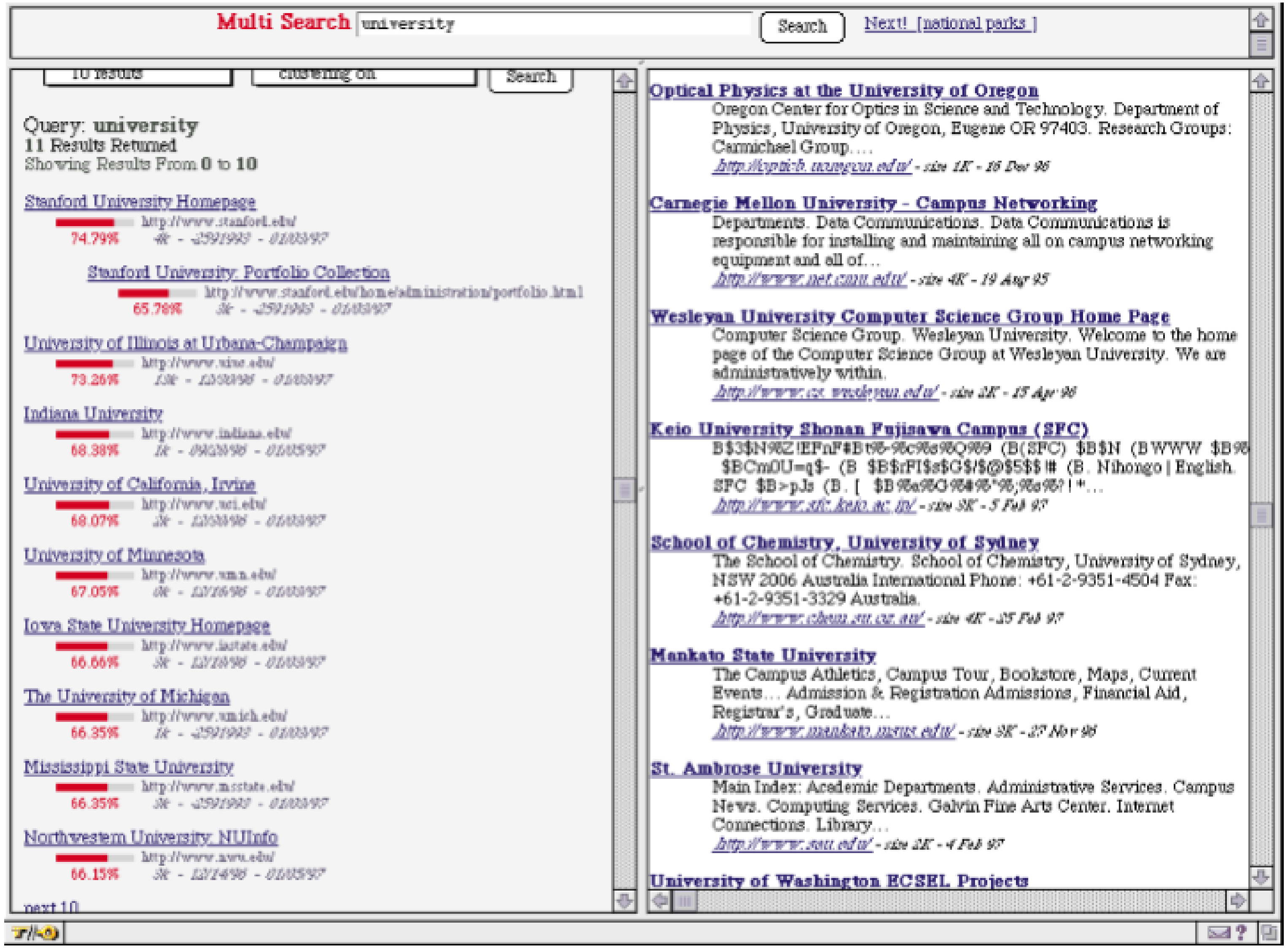
CPSC 340: Machine Learning and Data Mining

PageRank

Fall 2019

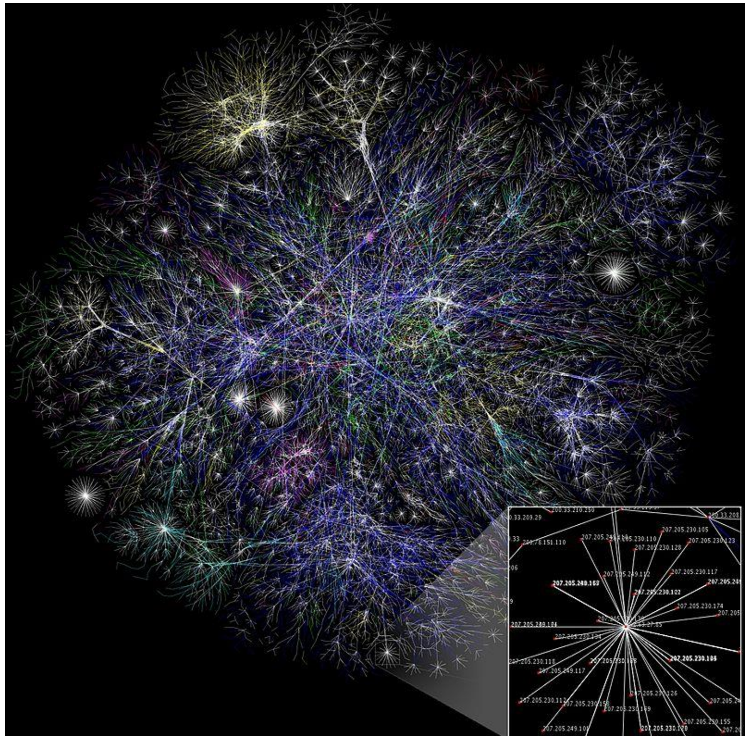
Web Search before Google



http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

Unsupervised Graph-Based Ranking

- We want to rank "importance" based on graph between examples.
  - Every webpage is a node, and every web-link is an edge.
  - Every paper is a node, and every citation is an edge.
  - Every Facebook user is a node, and every "friendship" is an edge.



https://en.wikipedia.org/wiki/Scale-free_network
http://blog.revolutionanalytics.com/2010/12/facebooks-social-network-graph.html
http://mathematica.stackexchange.com/questions/11673/how-to-play-with-facebook-data-inside-mathematica

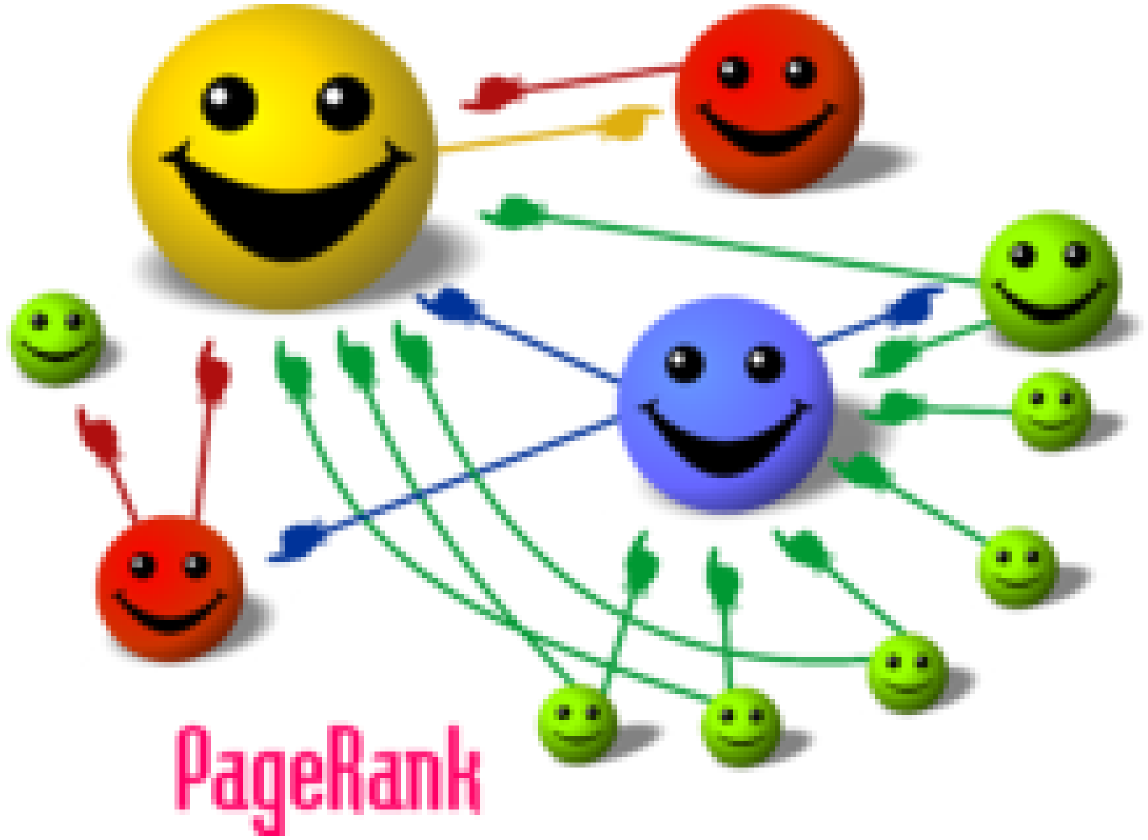
Unsupervised Graph-Based Ranking

- We want to rank "importance" based on graph between examples.
  - Every webpage is a node, and every web-link is an edge.
  - Every paper is a node, and every citation is an edge.
  - Every Facebook user is a node, and every "friendship" is an edge.

- Key idea: use links (edges) to predict importance of nodes.
- Many link analysis methods, usually with recursive definitions:
  - A journal is "influential" if it is cited by "influential" journals.
- We will discuss PageRank, Google's original ranking algorithm.

PageRank

- Wikipedia's cartoon illustration of PageRank:
  - Large face => higher rank.



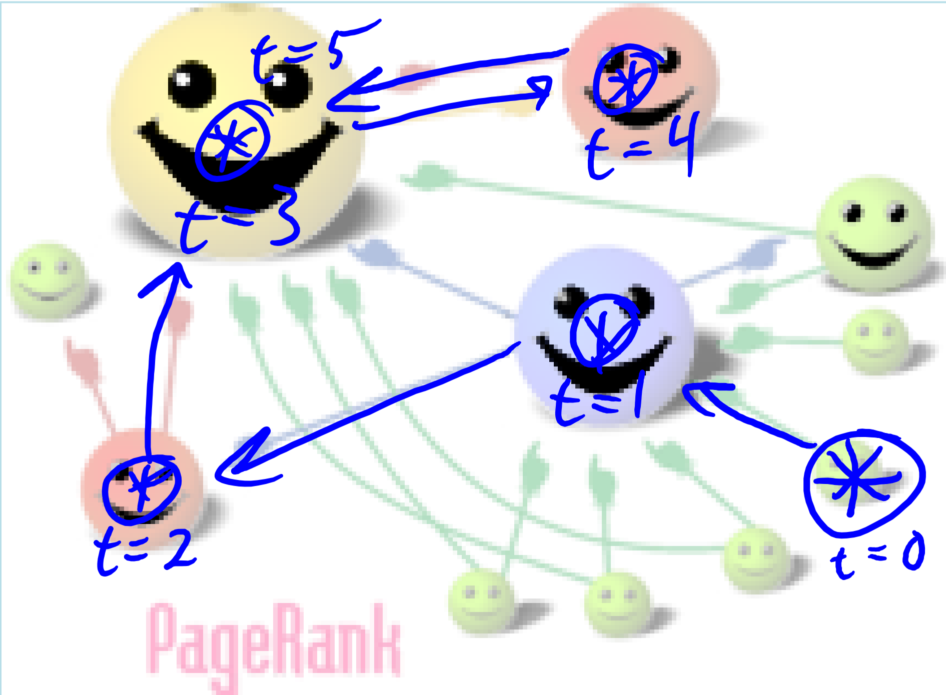
- Key ideas:
  - Important webpages are linked from other important webpages.
  - Link is more meaningful if a webpage has few links.

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

Random Walk View of PageRank

- PageRank algorithm can be interpreted as a random walk:
  - At time t=0, start at a random webpage.
  - At time t=1, follow a random link on the current page.
  - At time t=2, follow a random link on the current page.

    ....
- PageRank:
  - Probability of landing on page as t->∞.
- Obvious problem:
  - Pages with no in-links have a rank of 0.
  - Algorithm can get "stuck" in part of the graph.



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

Random Walk View of PageRank

- Fix: add small probability of going to a random webpage at time ‘t’.
- Damped PageRank algorithm:
  - At time t=0, start at a random webpage.
  - At time t=1:
    - With probability α (like 10%): go to a random webpage.
    - With probability (1- α): follow a random link on the current page.
  - At time t=2, follow a random link on the current page.
    - With probability α: go to a random webpage.
    - With probability (1- α): follow a random link on the current page.
- PageRank:
  - Probability of landing on page as t->∞.

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

PageRank Computation

- "Monte Carlo" method for computing PageRank:
  - Just run the random walk algorithm a really long time.
  - Count the number of times you visit each webpage.
    - Maybe include a "burn in" time at the start where you don't count pages.
    - Can parallelize by using 'm' independent surfers.
  - Intuitive but slow.
- It can also be solved analytically with SVD:
  - But $O(n^3)$ for 'n' webpages.
- Google's approach is the power method:
  - Repeated multiplication by transition matrix: O(nLinks) per iteration.

Application: Game of Thrones

- PageRank can be used for other applications.
- “Who is the main character in the Game of Thrones books?”

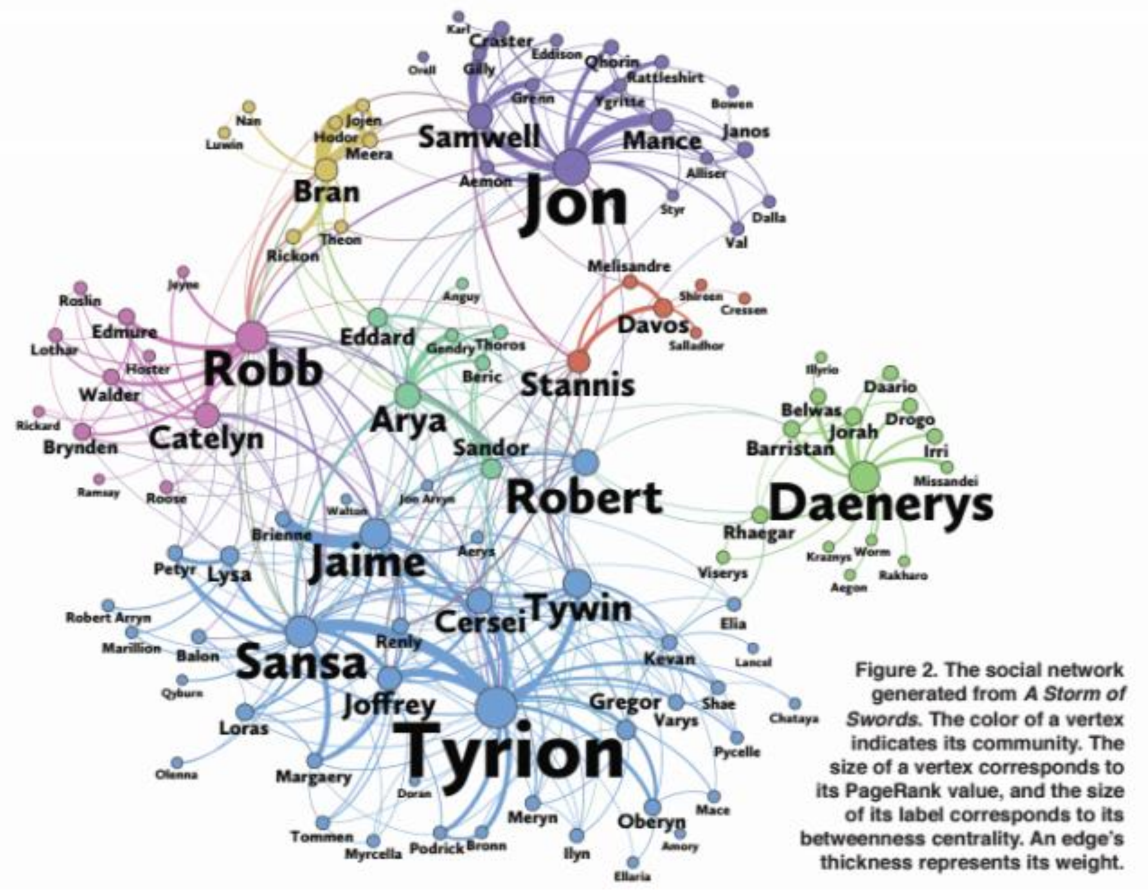


Figure 2. The social network generated from *A Storm of Swords*. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

http://qz.com/650796/mathematicians-mapped-out-every-game-of-thrones-relationship-to-find-the-main-character

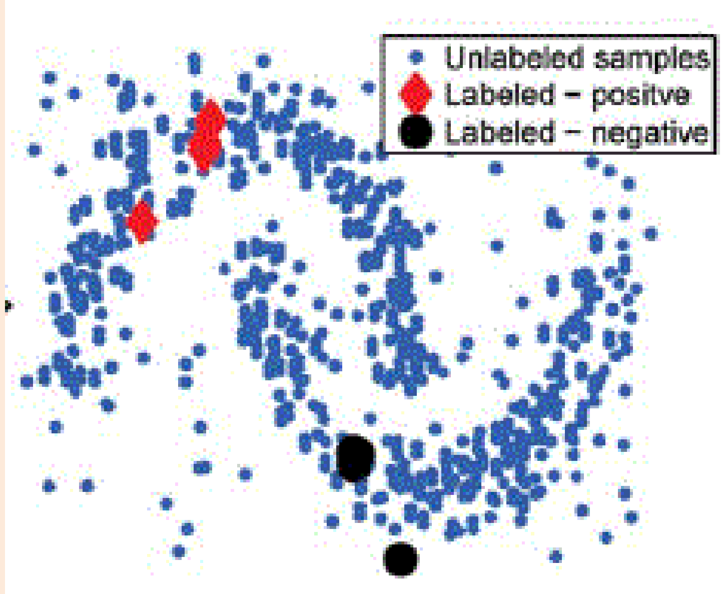
Ranking Discussion

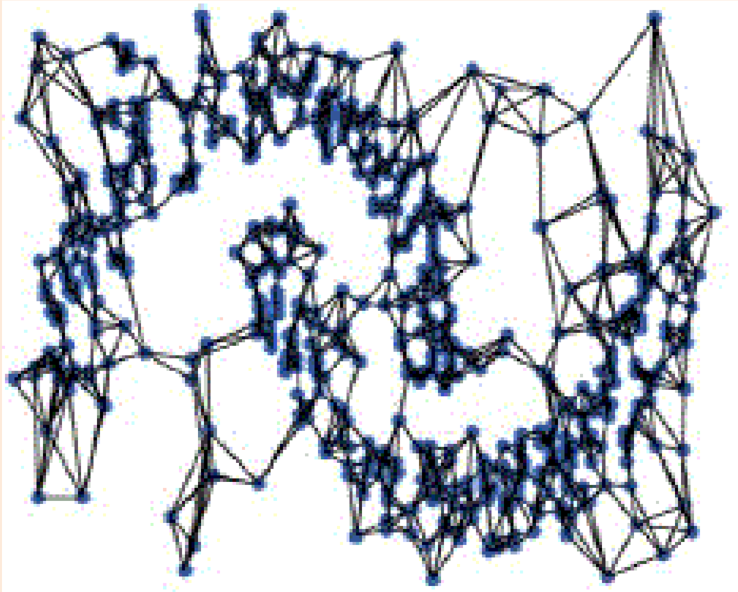
- Modern ranking methods are more advanced:
  - Guarding against methods that exploit algorithm.
  - Removing offensive/illegal content.
  - Supervised and personalized ranking methods.
  - Take into account that you often only care about top rankings.
  - Also work on diversity of rankings:
    - E.g., divide objects into sub-topics and do weighted “covering” of topics.
  - Persistence/freshness as in recommender systems (news articles).

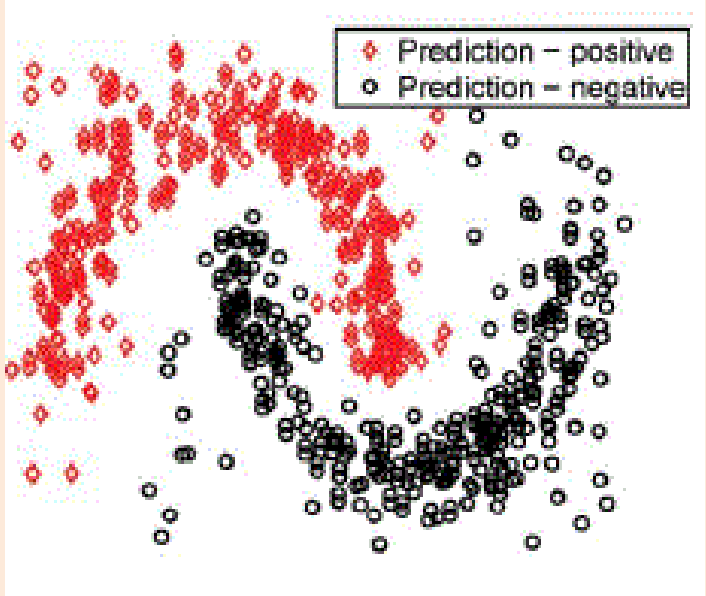
(pause)

Previously: Graph-Based Semi-Supervised Learning

- Graph-based semi-supervised learning:
  - Define weighted graph on training examples:
    - For example, use KNN graph or points within radius ‘ε’.
    - Weight is how ‘important’ it is for nodes to share label.



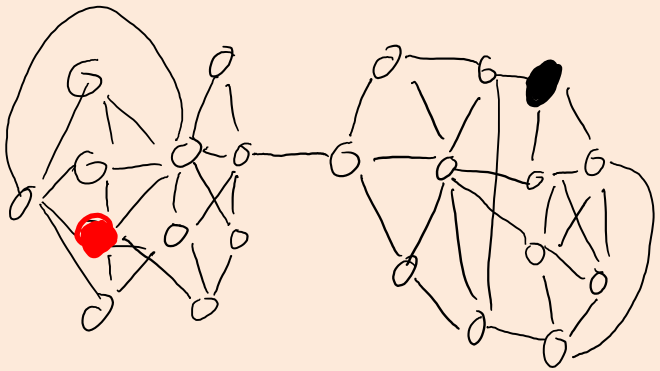




http://www.ee.columbia.edu/ln/dvmm/pubs/publications.html

PageRank, Label Propagation, and Random Walks

- Standard graph-based SSL also has a random walk interpretation:
  - At time t = 0, set your state to the node you want to label.
  - At time t > 0, move to a random neighbor.
    - With probability proportional to w_{ij} (how much we want them to be similar).
  - If you land on a labeled node, choose that label for this "round".
- Final predictions are probabilities of outputting each label.



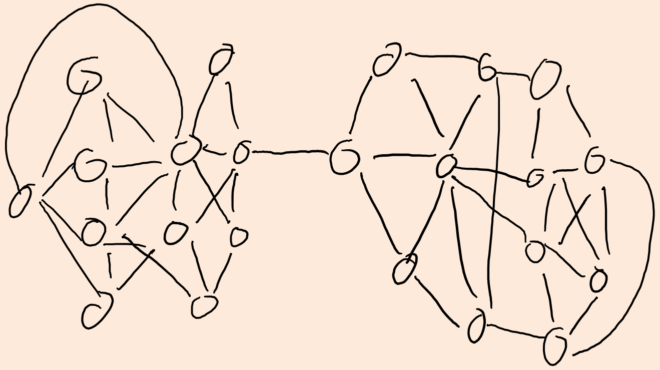
https://en.wikipedia.org/wiki/PageRank

What else can we do with random walks?

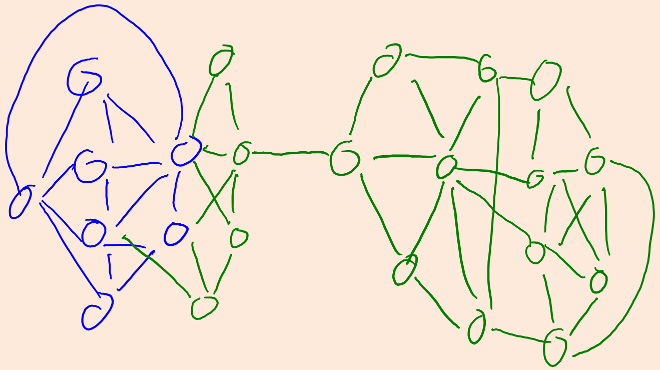
- We’ve discussed random walks for ranking and SSL.
  - Useful for problems defined on graphs.
  - We can convert from features to graphs using things like KNN graphs.
- Random walks for other tasks:
  - Outlier detection with outrank:
    - Examples with low PageRank are considered outliers (can detect outlier clusters).

What else can we do with random walks?

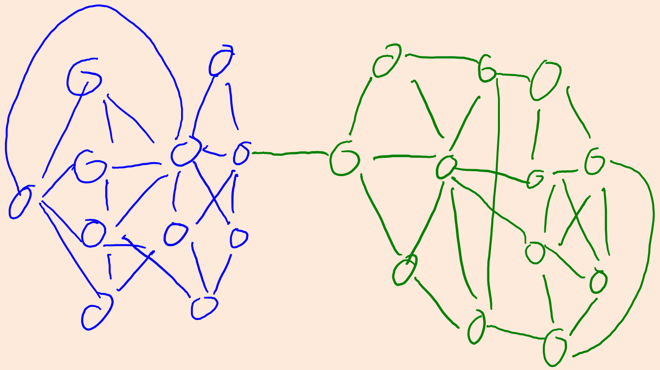
- We've discussed random walks for ranking and SSL.
  - Useful for problems defined on graphs.
  - We can convert from features to graphs using things like KNN graphs.
- Random walks for other tasks:
  - Clustering with spectal clustering (and "spectral graph theory):
    - "If we start in cluster 'c', random walk should tend to stay in cluster 'c'".



Graph representation of data

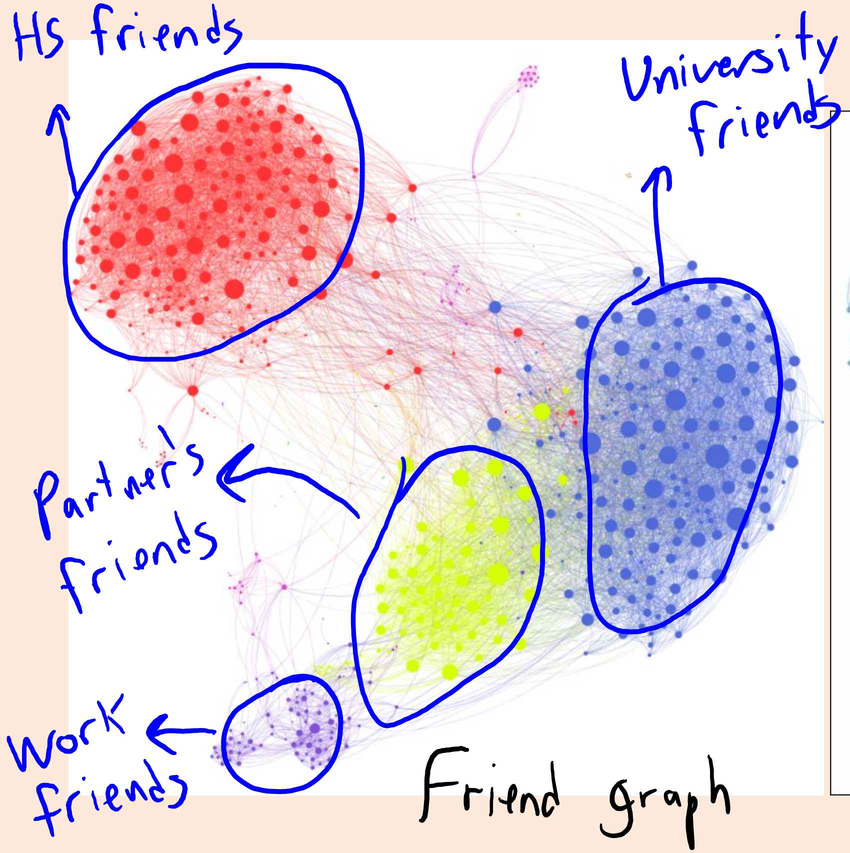


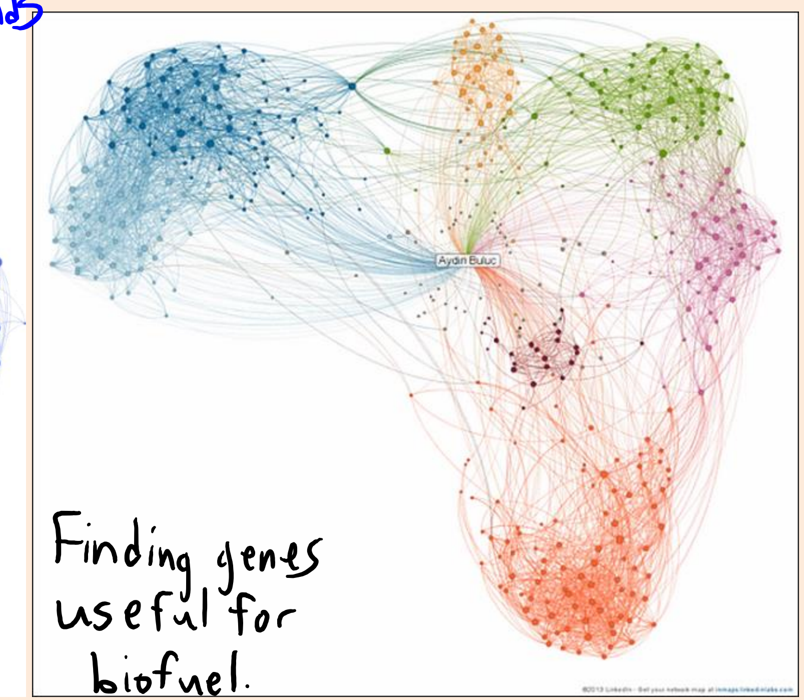
Bad clustering

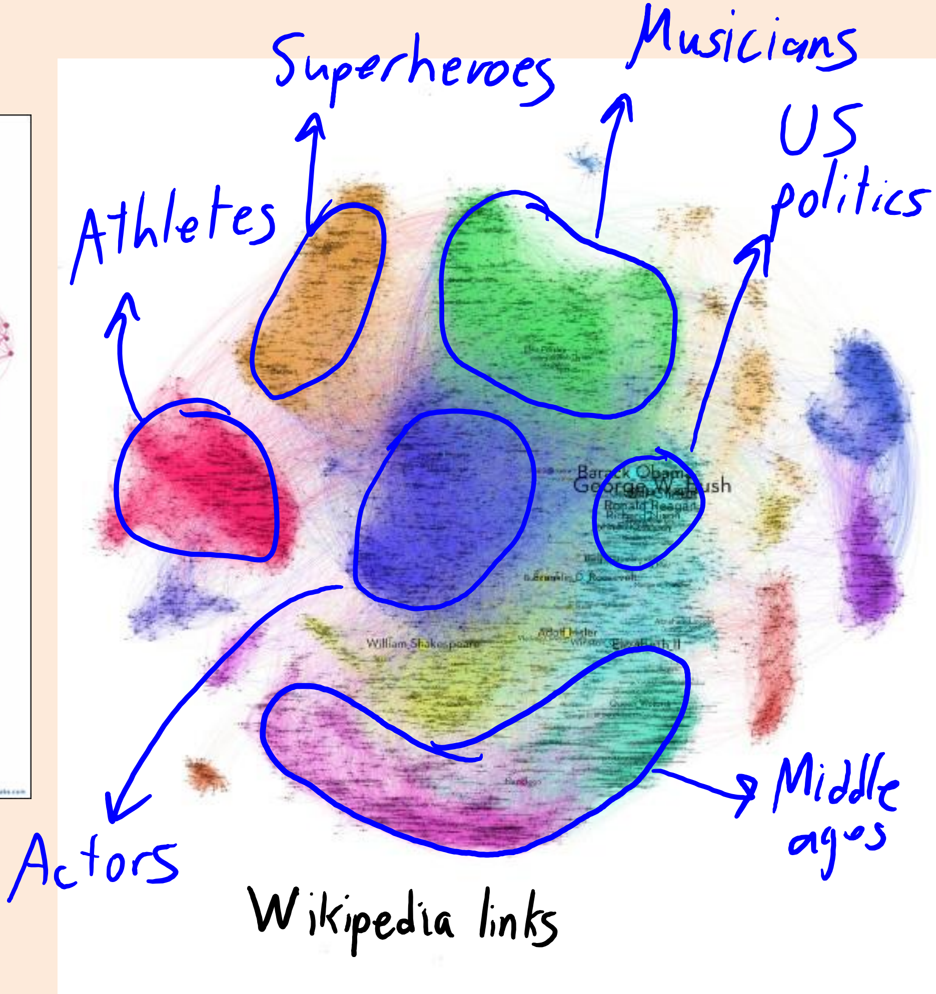


Good clustering.

Graph-Based Clustering Methods



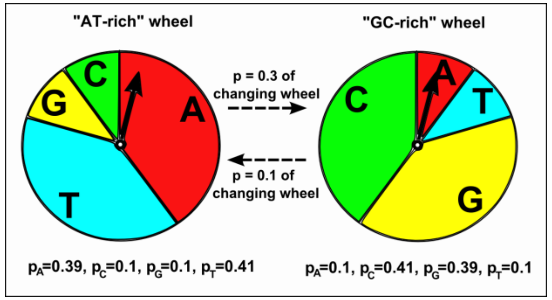


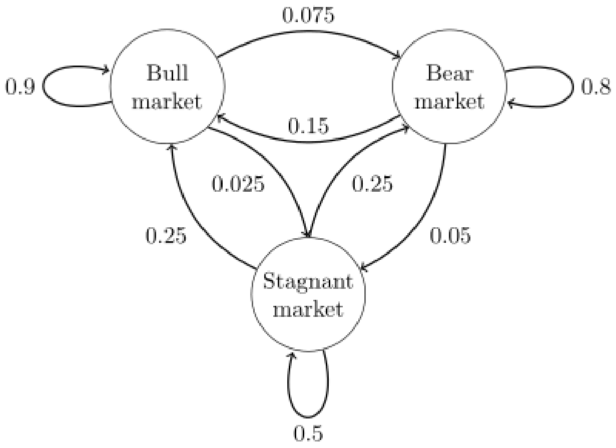


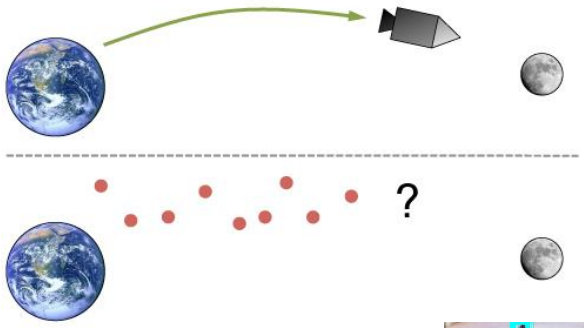
https://griffsgraphs.wordpress.com/tag/clustering/
http://ascr-discovery.science.doe.gov/2013/09/sifting-genomes/
https://www.hackdiary.com/2012/04/05/extracting-a-social-graph-from-wikipedia-people-pages/

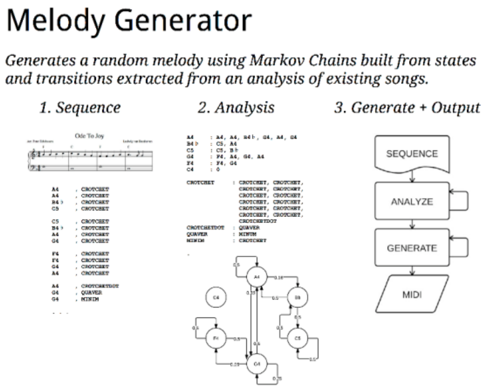
Markov Chains

- These random walk algorithms are special cases of Markov chains:
  - Most common framework for modeling sequences.
    - Bioinformatics, physics/chemistry, speech recognition, predator-prey models, language tagging/generation, computing integrals, economic models, flying airplanes, tracking missiles/players, modeling music.









https://en.wikipedia.org/wiki/Markov_chain
http://www.cs.uml.edu/ecg/index.php/AIfall11/MarkovMelodyGenerator
http://a-little-book-of-r-for-bioinformatics.readthedocs.org/en/latest/src/chapter10.html
https://plus.maths.org/content/understanding-unseen
http://www.cs.ubc.ca/~okumak/research.html

(pause)

Example: Vancouver Rain Data

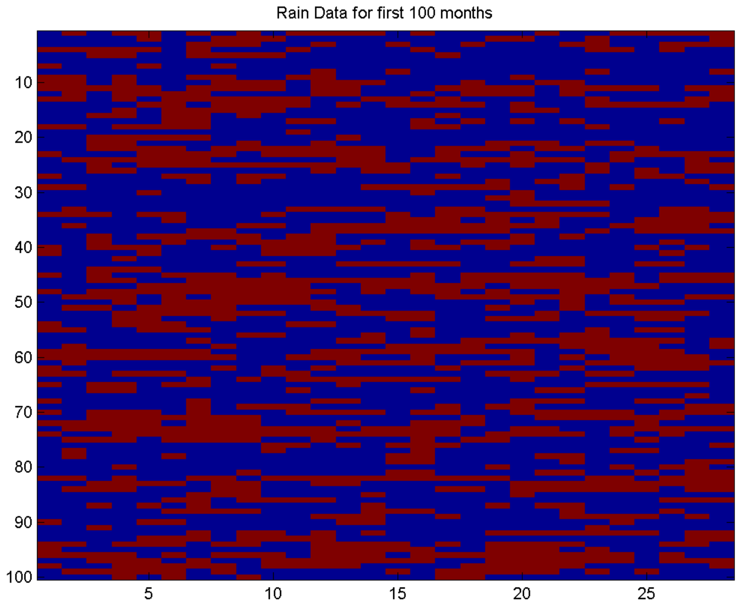
- Consider modeling the "Vancouver rain" dataset.

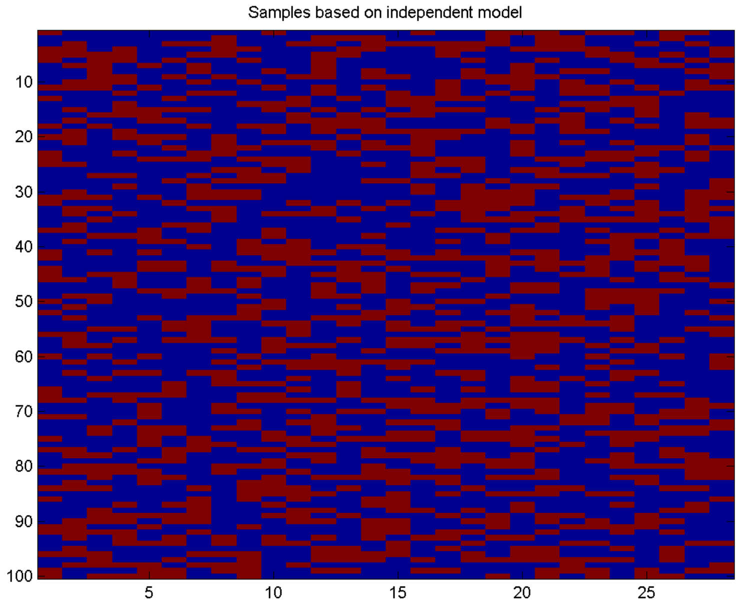
	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	…
Month 1	0	0	0	1	1	0	0	1	1	
Month 2	1	0	0	0	0	0	1	0	0	
Month 3	1	1	1	1	1	1	1	1	1	
Month 4	1	1	1	1	0	0	1	1	1	
Month 5	0	0	0	0	1	1	0	0	0	
Month 6	0	1	1	0	0	0	0	1	1	

- A time-series dataset where $x_t = 1$ if it rained on day 't'.
- The strongest signal in the data is the simple relationship:
  - If it rained yesterday, it's likely to rain today (> 50% chance that $x_{t-1} = x_t$).

Example: Vancouver Rain Data

- If we assume x_t are independent, we get $p(x_t = 1) = 0.41$ (sadly).
  - Real data vs. samples from independent Bernoulli model:

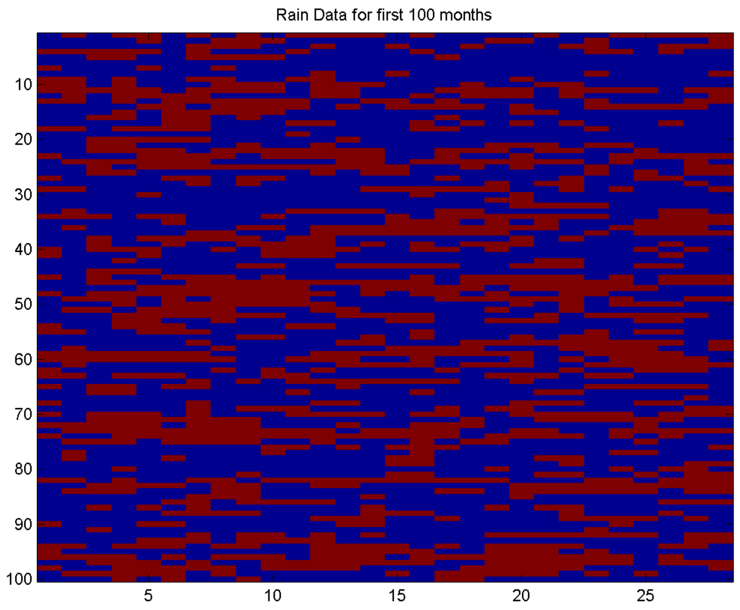


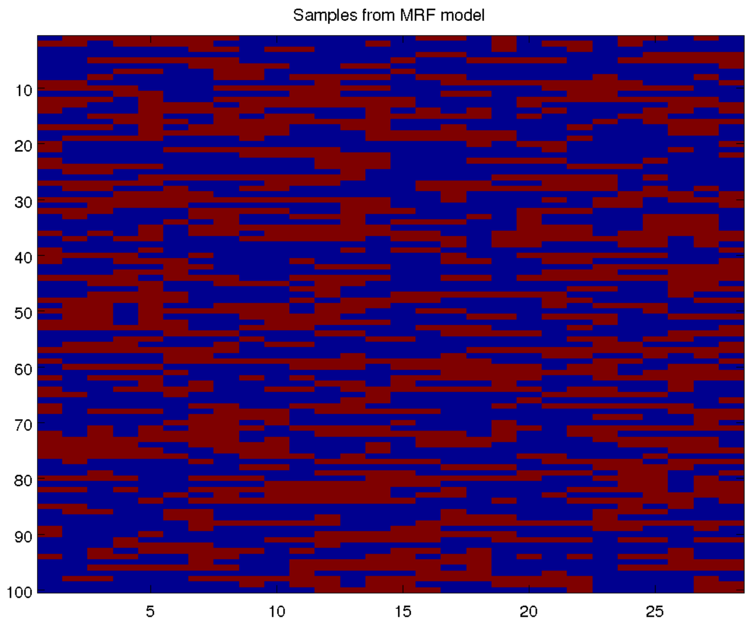


  - Making days independent misses correlation.

Markov Chain Model of Rain Data

- A better model for the rain data is a Markov chain:
  - Captures dependency of x_t on x_{t-1}.





  - We model $p(x_t \mid x_{t-1})$: probability of rain today given yesterday's value.

Markov Chain Ingredients (MEMORIZE)

- Markov chain ingredients:
  - State space:
    - Set of possible states (indexed by ‘s’) we can be in at time ‘t’ (“rain” or “not rain”).
  - Initial probabilities:
    - $p(x_1 = s)$ that we start in state ‘s’ at time 1.
  - Transition probabilities:
    - $p(x_t = s \mid x_{t-1} = s')$ that we move to state s from state s’ at time ‘t’.
      - Probability that it rains today, given what happened yesterday.

- For PageRank: each webpage is a state ‘s’.
  - Initial probability is random.
  - Go to random page with probability α, otherwise go to random neighbour.

Markov Chain Probability and Markov Property

- Markov chain probability for a sequence $x_1, x_2, \dots, x_d$:

$$p(x_1, x_2, \dots, x_d) = p(x_1)p(x_2 | x_1)p(x_3|x_2) \cdots p(x_d | x_{d-1})$$

- This assumes the Markov property:

$$p(x_t | x_1, x_2, x_3, \dots, x_{t-1}) = p(x_t | x_{t-1})$$

  – That x_t is independent of the past given x_{t-1}.
    - To predict "rain", we only need to know whether it rained yesterday.

Markov Chain Applications

- 9 Applications
  - 9.1 Physics
  - 9.2 Chemistry
  - 9.3 Testing
  - 9.4 Speech Recognition
  - 9.5 Information sciences
  - 9.6 Queueing theory
  - 9.7 Internet applications
  - 9.8 Statistics
  - 9.9 Economics and finance
  - 9.10 Social sciences
  - 9.11 Mathematical biology
  - 9.12 Genetics
  - 9.13 Games
  - 9.14 Music
  - 9.15 Baseball
  - 9.16 Markov text generators

Homogeneous Markov Chains

- We usually assume that the Markov chain is homogeneous:
  - Transition probabilities $p(x_t = s \mid x_{t-1} = s')$ are same for all 't'.

- Given 'n' samples, MLE for homogeneous Markov chain is:

Initial: $p(x_1 = s) = \frac{\text{number of times we start in state } s}{n}$

Transition: $p(x_t = s \mid x_{t-1} = s') = \frac{\text{number of times we went from } s' \text{ to } s}{\text{number of times we went from } s' \text{ to anything}}$

- So given one or more sequences, learning is just counting.
  - Like in naïve Bayes.

Computation with Markov Chains

- Common things we do with Markov chains:
  - Sampling: generate sequences that follow the probability.
    - This is what our "random walk" algorithms are doing.
  - Inference: compute probability of being in state 's' at time 't'.
  - Decoding: compute most likely sequence of states.
  - Conditioning: do any of the above, assuming $x_t = s$ for some 't' and 's'.
    - For example, "filling in" missing parts of a sequence.
  - Stationary distribution: probability of being 's' at 't' goes to ∞.
    - PageRank.

Fun with Markov Chains

- Markov chains “explained visually”:
  - http://setosa.io/ev/markov-chains
- Snakes and ladders:
  - http://datagenetics.com/blog/november12011/index.html
- Candyland:
  - http://www.datagenetics.com/blog/december12011/index.html
- Yahtzee:
  - http://www.datagenetics.com/blog/january42012
- Chess pieces returning home and K-pop vs. ska:
  - https://www.youtube.com/watch?v=63HHmjlh794

Application: Voice Photoshop

- [Adobe VoCo](http://gfx.cs.princeton.edu/pubs/Jin_2017_VTI/Jin2017-VoCo-paper.pdf) uses decoding as part of synthesizing voices:

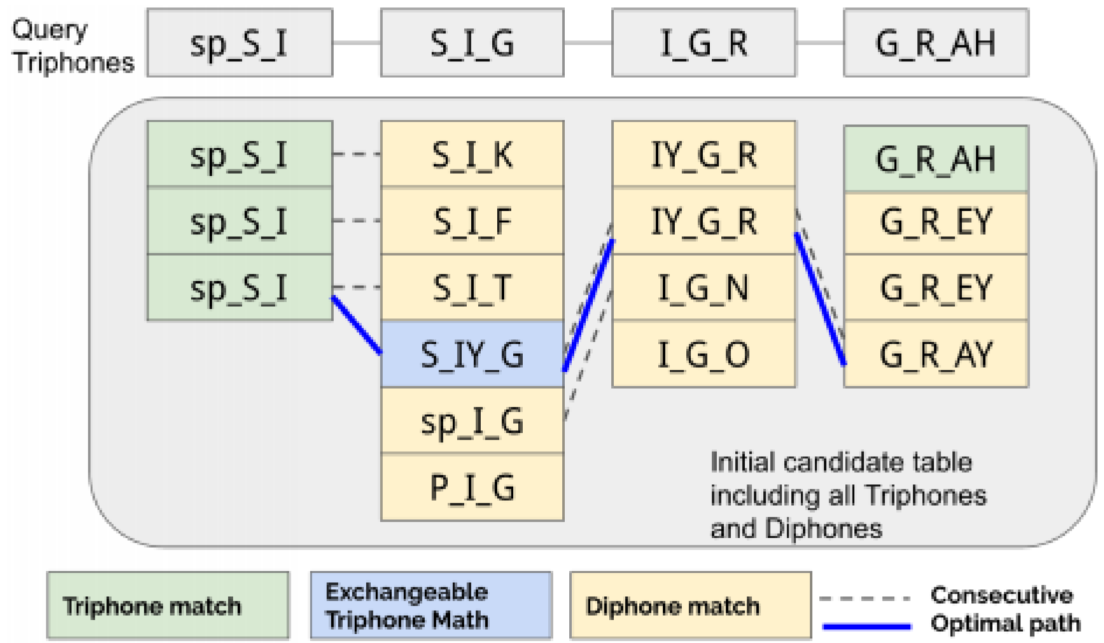


Fig. 7. Dynamic triphone preselection. For each query triphone (top) we find a candidate set of good potential matches (columns below). Good paths through this set minimize differences from the query, number and severity of breaks, and contextual mismatches between neighboring triphones.

http://gfx.cs.princeton.edu/pubs/Jin_2017_VTI/Jin2017-VoCo-paper.pdf

Summary

- Graph-based ranking uses links to solve ranking queries.
  - PageRank is based on a model of a random web user.
- Markov chains model dependency between states x_t across time.
  - Based on Markov assumption: “independence of past given last time”.