CPSC 340: Machine Learning and Data Mining

Ranking

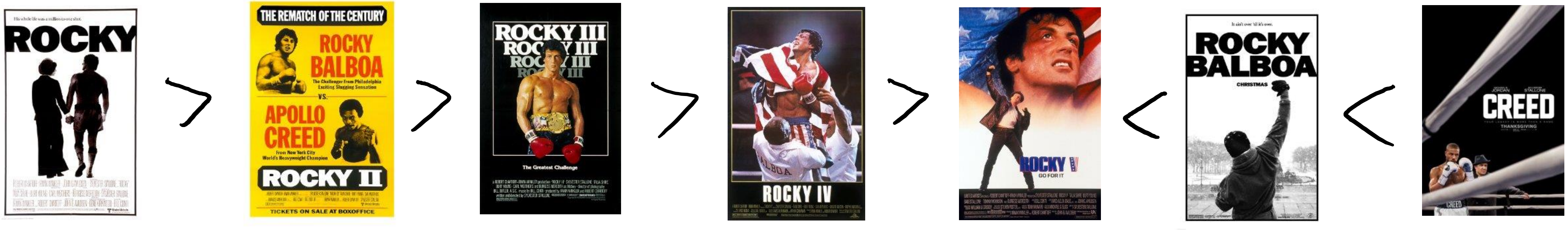
Fall 2016

Admin

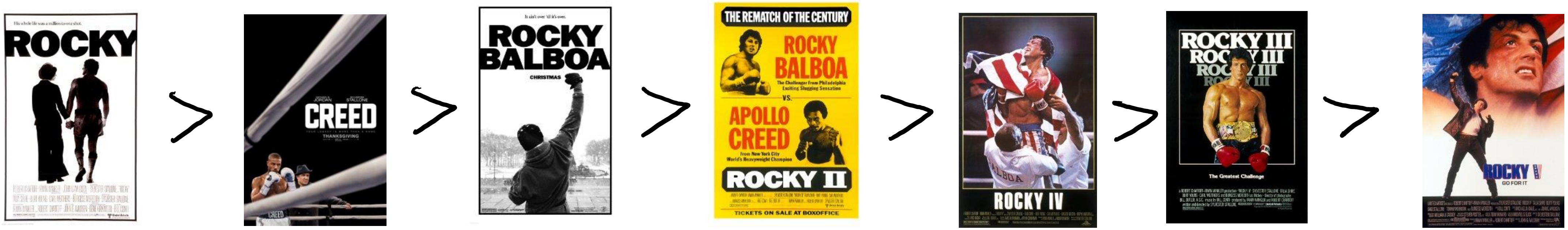
- Assignment 5:
  - 2 late days to hand in Wednesday, 3 for Friday.
- Assignment 6:
  - Due Friday, 1 late day to hand in next Monday, etc.
- Final:
  - December 12 (8:30am – HEBB 100)
  - Covers Assignments 1-6.
  - List of topics posted.
  - Final from last year will be posted Friday.
  - Closed-book, cheat sheet: 4-pages each double-sided.

Ranking

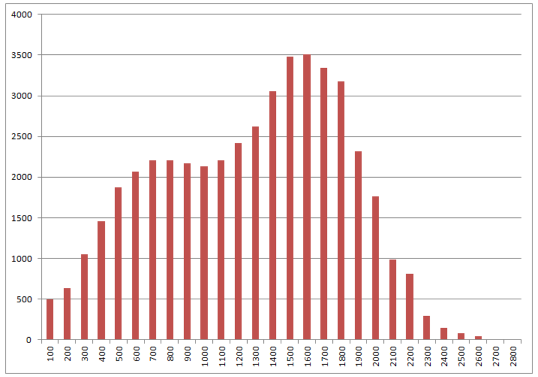
- The ranking problem:
  - Input: a set of objects and some information about “ordering”.



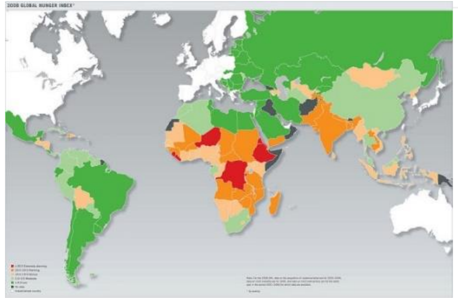
  - Output: an ordering of the objects.

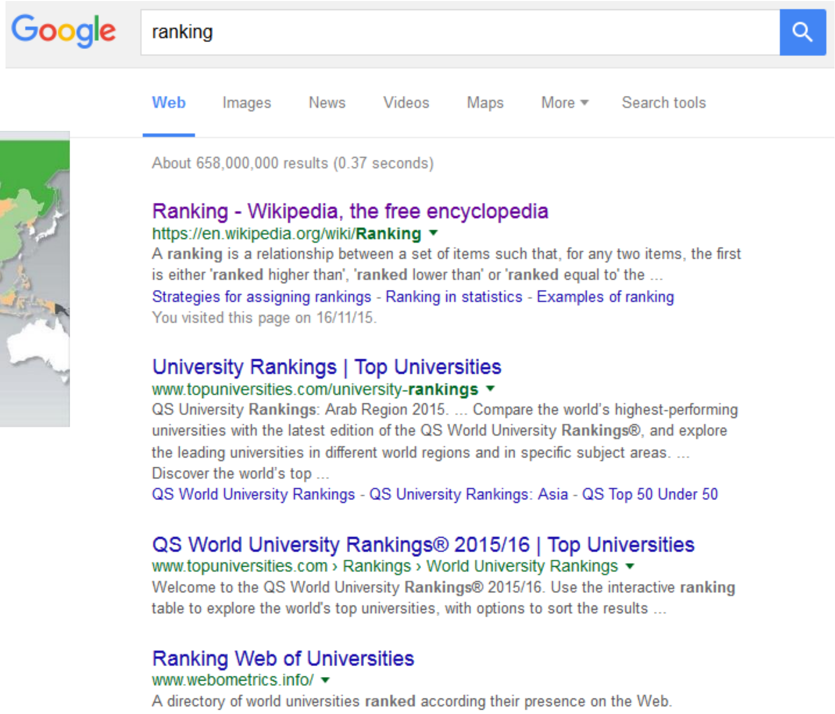


Ranking



- The ranking problem:
  - Input: a set of objects and some measure of relative “ordering”.
  - Output: an ordering of the objects.
- Examples:
  - Country comparisons (Global Hunger Index).
  - Academic journals (Impact factor).
  - Sports/gaming (Elo and TrueSkill).
  - Internet search engines.
- Large, diverse, and well-studied topic.
  - We focus on learning to rank.





http://chess.stackexchange.com/questions/2550/whats-the-average-elo-rating-whats-the-average-uscf-rating
https://commons.wikimedia.org/wiki/File:GHI_2008_map.jpg

Supervised Ranking

- Ranking based on supervised learning:
  – We have features x_i for each object 'i', and "information" about labels y_i.
- Forms the "information" can take:
  1. Item relevance:
     - Explicit numerical "scores" y_i.
  2. Pairwise preference:
     - Pairs 'i' and 'j' where we know $y_i > y_j$.
     - But we don't know the "score" of any items.
  3. Total/partial ordering:
     - Larger sets of items where we know $y_i > y_j > y_k > y_m > \ldots$

Supervised Ranking with Item Relevance

- With item relevance we have explicit "score" y_i for each object 'i'.
- We can rank with regression:

$$\text{Find 'w' minimizing } f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2$$

- Compute score of new object 'i' based on its features 'x_i'.

$$\hat{y}_i = w^Tx_i$$

- If scores are ordinal {1,2,3,..,k}, can use ordinal logistic regression.

Supervised Ranking with Query

- Common variation on ranking includes query 'q'.
  - E.g., for web search it could the keywords.
- Can adapt item relevance to this setting:
  - Measure features x_{iq} of object/query combination.
  - Item relevance y_{iq} gives "score" of object/query combination.

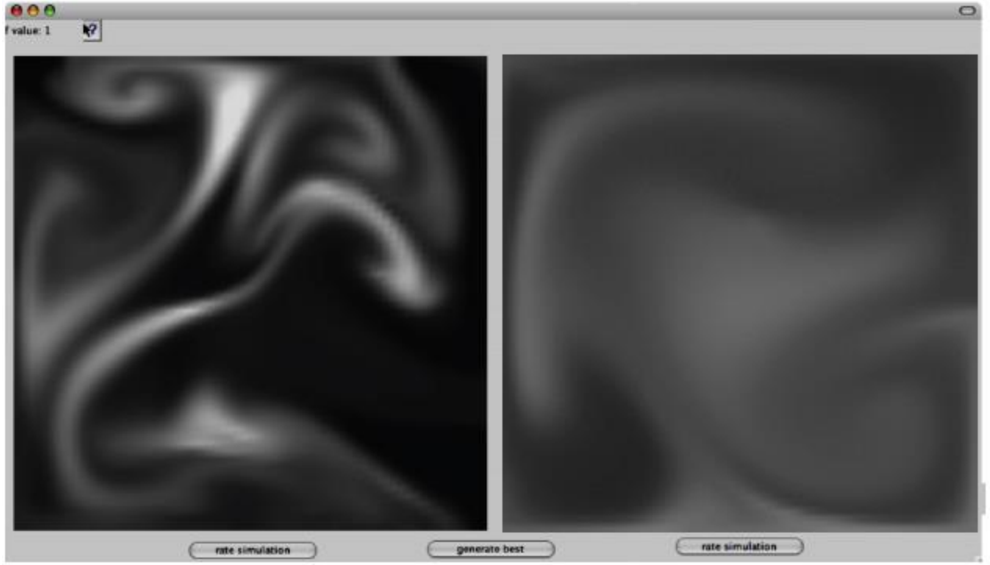
find 'w' minimizing $f(w) = \frac{1}{2} \sum_{(i,q) \in R} (w^T x_{iq} - y_{iq})^2$

set of available ratings

- Unfortunately, item relevance may be hard to get:
  - Active human effort to produce meaningful labels across queries/objects.
  - How do you compare 'CPSC 340' to 'shoe' or 'moon' to 'Tuesday' on same scale?

Supervised Ranking with Pairwise Preferences

- More realistic is pairwise preferences:
  - We aren't given any explicit y_i values.
  - Instead we're given list of objects (i,j) where $y_i > y_j$.
  - E.g., which one looks more like 'smoke':
    - Much easier than asking artist for score.



- Can we design a loss function with this label information?

https://circle.ubc.ca/bitstream/handle/2429/30519/ubc_2011_spring_brochu_eric.pdf?sequence=3

Digression: Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) = \frac{1}{1 + \exp(-y_i w^T x_i)} = \frac{\exp(\frac{1}{2} y_i w^T x_i)}{\underbrace{\exp(\frac{1}{2} y_i w^T x_i) + \exp(-\frac{1}{2} y_i w^T x_i)}_{\text{Same normalizing constant for } y_i = +1 \text{ and } x_i = -1}} \propto \exp(\tfrac{1}{2} y_i w^T x_i)$$

Digression: Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$p(y_i \mid x_i, w) \propto \exp(\frac{1}{2} y_i w^T x_i)$

To classify y_i correctly, it's sufficient to have $\underbrace{\frac{p(y_i \mid x_i, w)}{p(-y_i \mid x_i, w)}} \geq \beta$ for some 'β' > 1

Notice that normalizing constant doesn't matter:

$$\frac{\exp(\frac{1}{2} y_i w^T x_i)}{\exp(-\frac{1}{2} y_i w^T x_i)} \geq \beta$$

Digression: Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$p(y_i \mid x_i, w) \propto \exp(\frac{1}{2} y_i w^T x_i)$

We need: $\frac{\exp(\frac{1}{2} y_i w^T x_i)}{\exp(-\frac{1}{2} y_i w^T x_i)} \geq \beta$

Take log:

$\log\left(\frac{\exp(\frac{1}{2} y_i w^T x_i)}{\exp(-\frac{1}{2} y_i w^T x_i)}\right) \geq \log(\beta) \iff \frac{1}{2} y_i w^T x_i + \frac{1}{2} y_i w^T x_i \geq \log(\beta)$

$\Updownarrow$

$y_i w^T x_i \geq 1$ (if we choose $\log(\beta) = 1$)

Digression: Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) \propto \exp(\tfrac{1}{2} y_i w^T x_i)$$

We need: $\dfrac{\exp(\frac{1}{2} y_i w^T x_i)}{\exp(-\frac{1}{2} y_i w^T x_i)} \geq \beta$

Or equivalently:

$y_i w^T x_i \geq 1$ (for $\beta = \exp(1)$)

Define a loss function by amount of constraint violation:

$$\max\{\underbrace{0}_{\text{when } 1 - y_i w^T x_i \leq 0},\ \underbrace{1 - y_i w^T x_i}_{\text{when } 1 - y_i w^T x_i \geq 0}\}$$

We get SVMs by looking at regularized average loss:

$$f(w) = \sum_{i=1}^{n} \max\{0, 1 - y_i w^T x_i\} + \frac{\lambda}{2}\|w\|^2$$

Digression: Loss Functions from Probability Ratios

- General approach for defining losses using probability ratios:
  1. Define constraint based on probability ratios.
  2. Minimize violation of logarithm of constraint.
- Example: softmax => multi-class SVMs.

Assume: $p(y_i = c \mid x_i, w) \propto \exp(w_c^T x_i)$

Want: $\dfrac{p(y_i \mid x_i, w)}{p(y_i = c' \mid x_i, w)} \geq \beta$ for all c' and some $\beta > 1$

For $\beta = \exp(1)$ equivalent to

$$w_{y_i}^T x_i - w_{c'}^T x_i \geq 1$$

for all $c' \neq y_i$

Option 1: penalize all violations:

$$\sum_{c'=1}^{K} \max\{0, 1 - w_{y_i}^T x_i + w_{c'}^T x_i\}$$

Option 2: penalize only max violation:

$$\max_{c' \neq c} \{\max\{0, 1 - w_{y_i}^T x_i + w_{c'}^T x_i\}\}$$

Supervised Ranking with Pairwise Preferences

- Ranking with pairwise preferences:
  - We aren't given any explicit y_i values.
  - Instead we're given list of objects (i,j) where $y_i > y_j$.

Assume $p(y_i \mid X, w) \propto \exp(w^T x_i)$ is probability that object 'i' has highest rank.

Want: $\frac{p(y_i \mid X, w)}{p(y_j \mid X, w)} \geq \beta$ for all preferences (i,j)

For $\beta = \exp(1)$ equivalent to

$$w^T x_i - w^T x_j \geq 1$$

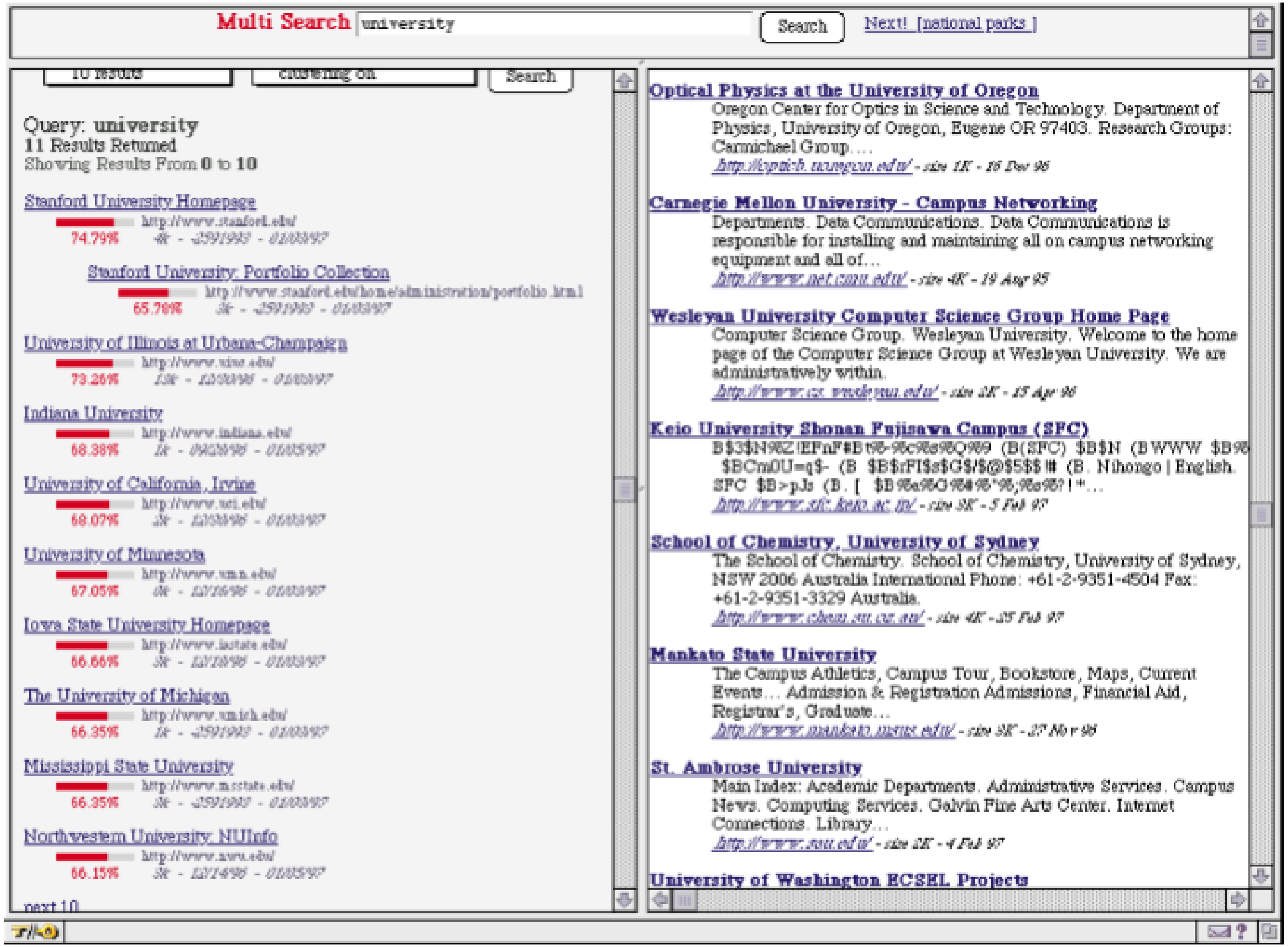
for preferences (i,j)

We can use $f(w) = \sum_{(i,j) \in R} \max\{0, 1 - w^T x_i + w^T x_j\}$

This approach can also be used to define losses for total/partial orderings. (but this information is hard to get)

(pause)

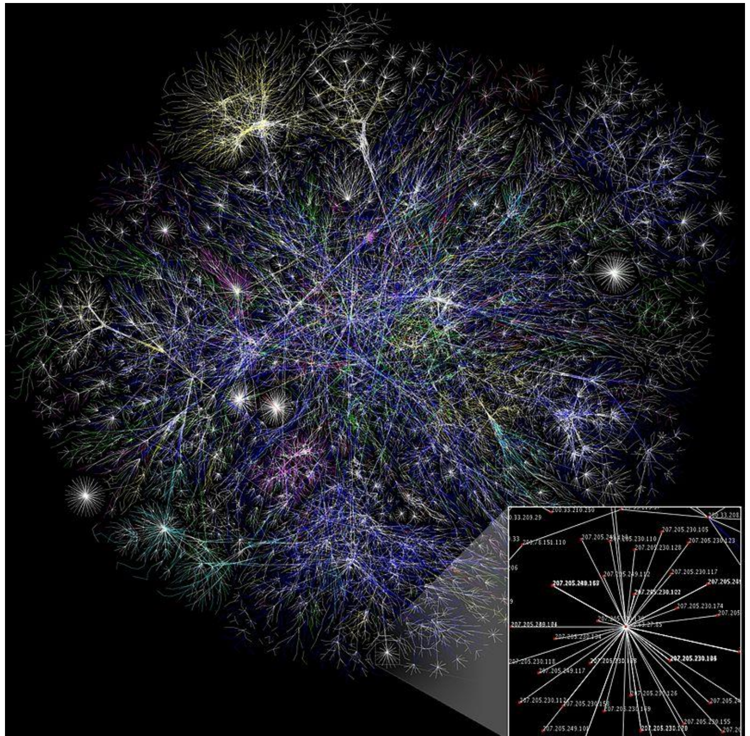
Web Search before Google



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

Unsupervised Graph-Based Ranking

- Instead of supervision, what if we have graph between examples?
  - Every paper is a node, and every citation is an edge.
  - Every Facebook user is a node, and every “friendship” is an edge.
  - Every webpage is a node, and every web-link 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

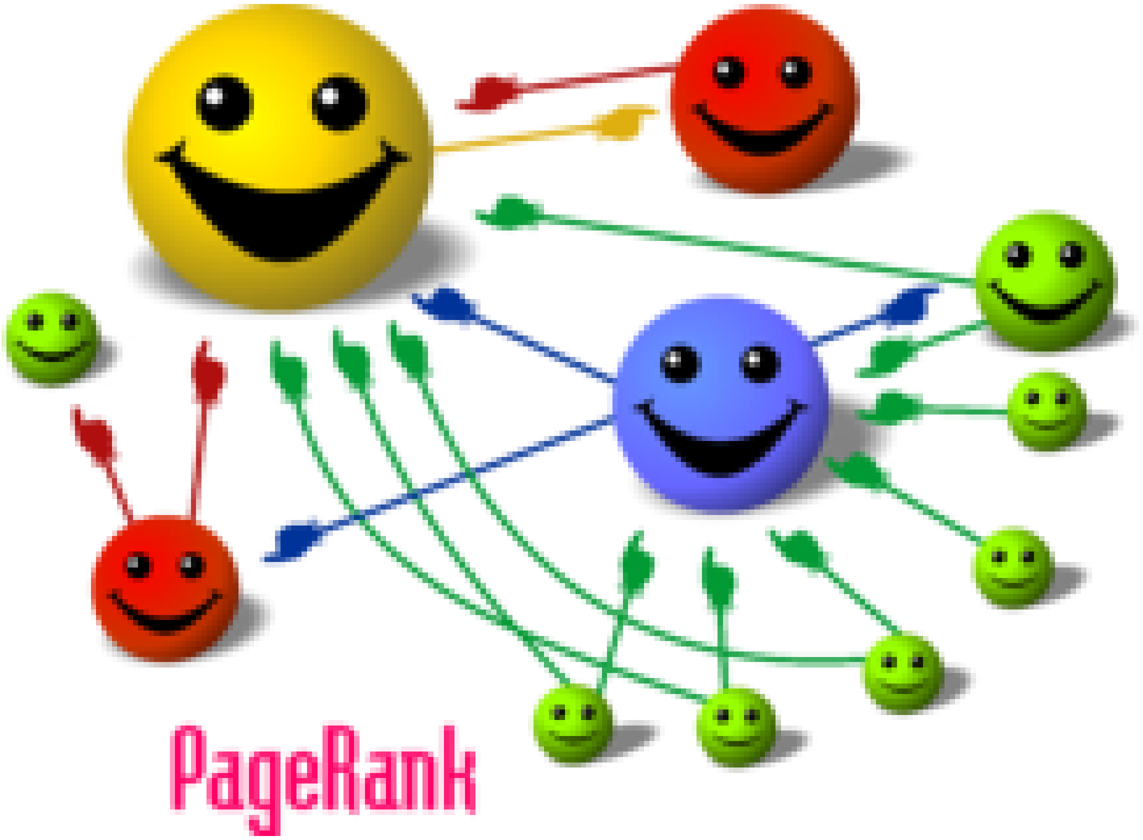
Unsupervised Graph-Based Ranking

- Instead of supervision, what if we have graph between examples?
  - Every paper is a node, and every citation is an edge.
  - Every Facebook user is a node, and every “friendship” is an edge.
  - Every webpage is a node, and every web-link is an edge.

- Key idea: use links (edges) to predict important 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.



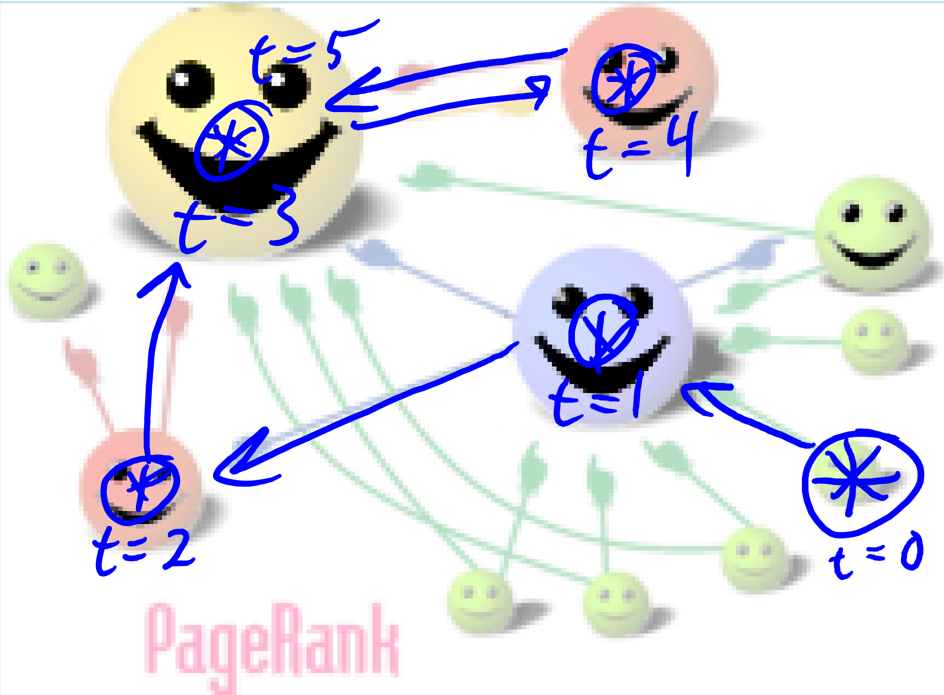
- Key ideas:
  - Important webpages are linked from other important webpages.
  - Link is more meaningful if a webpage has few links.

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.



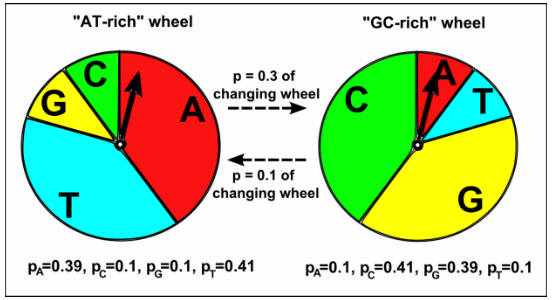
Random Walk View of PageRank

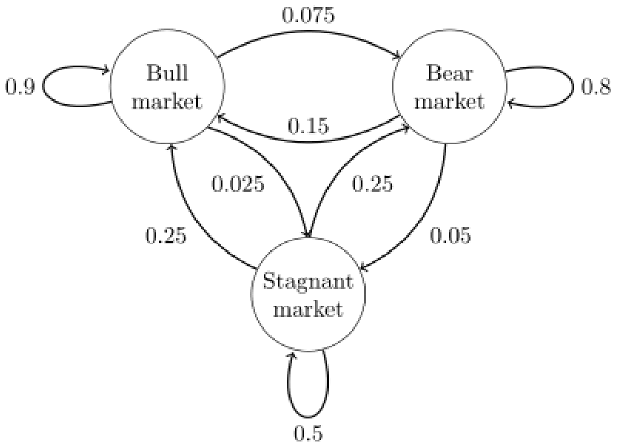
- Fix: add small probability of going to a random webpage at time ‘t’.
- Damped PageRank algorithm:
  - At time t=0, start at a random webpage.
  - At time t=1:
    - With probability α: go to a random webpage.
    - With probability (1- α): follow a random link on the current page.
  - At time t=2,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

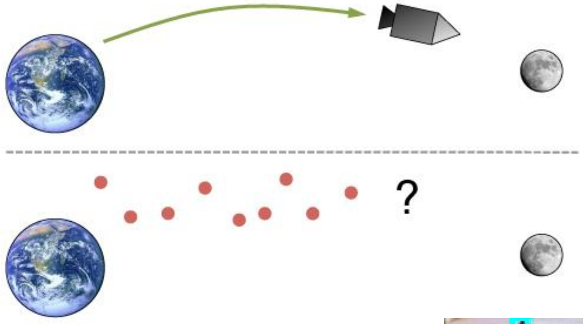
https://en.wikipedia.org/wiki/PageRank

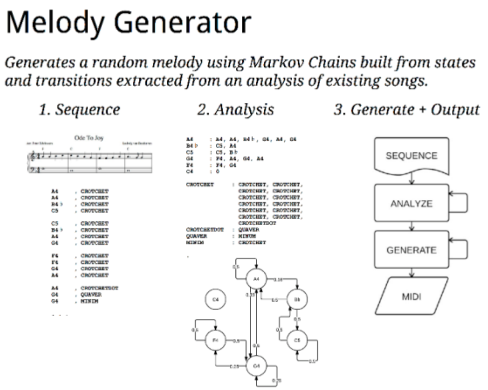
Markov Chains

- This random walk algorithm is a special case of a Markov chain:
  - Most common framework for modeling sequences.
    - Bioinformatics, physics/chemistry, speech recognition, predator-prey models, language tagging/generation, computing integrals, economic models, 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

Part 6: Markov Chains

- Random walk algorithm is a special case of a Markov chain.
- Markov chain ingredients:
  - State space:
    - Set of possible states we can be in at time ‘t’ (webpages for PageRank).
  - Initial probabilities:
    - $p(x_0 = s)$ that we start in state ‘s’ at time 0.
  - Transition probabilities:
    - $p(x_t = s \mid x_{t-1} = s')$ that we move to state s to state s’.
- This model makes the Markov assumption:
  - Our state time at ‘t’ only depends on the state at time t-1.
- Often assume homogeneous chain: transitions constant with ‘t’.

Markov Chains

- 3 things you can do with Markov chains:
  - You can simulate sequences:
    - Sample state x_0 from initial probabilities.
    - For t = 1:d
      - Sample x_t from transition probabilities.
  - Compute marginal probability of being in state ‘s’ at time ‘t’:
    - At time 0, just use the initial probabilities.
    - At time t > 0, marginalization and product rules gives recursive formula:

$$\underbrace{p(x_t = s)}_{\text{"marginal"}} = \sum_{s'=1}^{|S|} p(x_t = s, x_{t-1} = s') = \sum_{s'=1}^{|S|} \underbrace{p(x_t = s \mid x_{t-1} = s')}_{\text{transition probability}} \underbrace{p(x_{t-1} = s')}_{\text{"marginal" at } t-1}$$

matrix multiplicati

  - Compute stationary distribution (PageRank):
    - $P(x_t = s)$ as ‘t’ goes to infinity.

Exists and is unique under certain assumptions. $\alpha > 0$ works

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 random ‘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?”

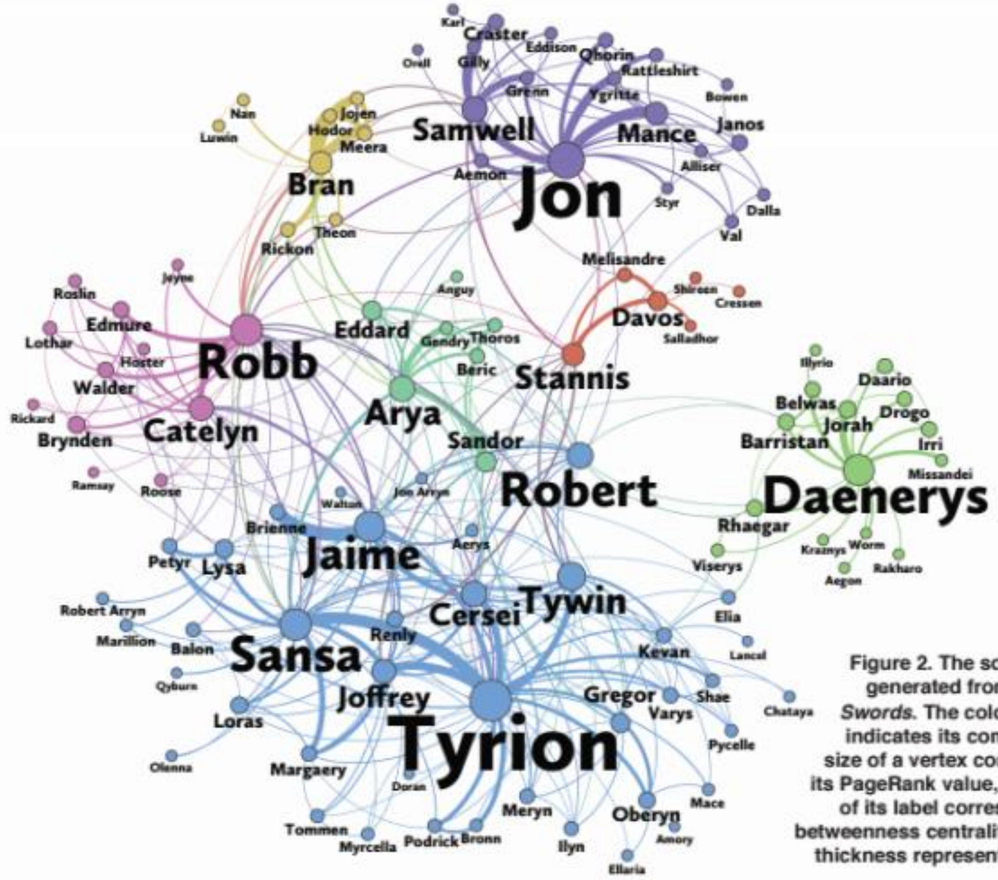


Figure 2. The social network generated from *A Storm of Swords*. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

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.
  - Personalized recommendations.
  - Take into account that you often only care about top rankings.
  - Define losses that are not additive across ratings.
    - "Precision at k": if we return k documents, how many are relevant?
    - "Average precision": precision at k averaged across values of 'k'.
  - You can still define losses based on probability ratios:
    - But you get exponential number of terms, need more advanced optimization tricks.
  - 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.

Summary

- Ranking orders objects based information about relationships.
- Supervised ranking contains explicit label information:
  - Item relevance assumes we have “scores” y_i.
  - Pairwise preferences assume we have relative rankings $y_i > y_j$.
- Probability ratios allow us to define more loss functions.
- Graph-based ranking uses links to solve ranking queries.
  - PageRank is based on a model of a random web user.
- Markov chains are a general framework for modeling sequences.

- Next time: finding all the cat videos on YouTube.