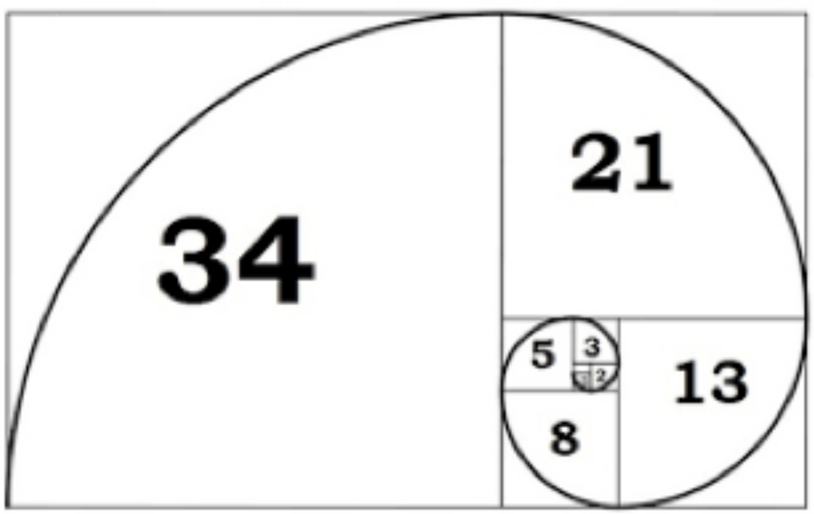
Sequences, Choices, and their Dynamics

Ravi Kumar
Google

Sequences, sequences, ...

AAGTCAAGCTGCTCTGTGGGCTGTGATCTGCCTCAAACCCACAGCCTGGGTAGCAGG
AGGACCTTGATGCTCCTGGCACAGATGAGGAGAATCTCTCTTTTCTCCTGCTTGAAG
GACAGACATGACTTTGGATTTCCCCAGGAGGAGTTTGGCAACCAGTTCCAAAAGGCT
GAAACCATCCCTGTCCTCCATGAGATGATCCAGCAGATCTTCAATCTCTTCAGCACA
AAGGACTCATCTGCTGCTTGGGATGAGACCCTCCTAGACAAATTCTACACTGAACTC
TACCAGCAGCTGAATGACCTGGAAGCCTGTGTGATACAGGGGGTGGGGGTGACAGAG
ACTCCCCTGATGAAGGAGGACTCCATTCTGGCTGTGAGGAAATACTTCCAAAGAATC
ACTCTCTATCTGAAAGAGAAGAAATACAGCCCTTGTGCCTGGGAGGTTGTCAGAGCA
GAAATCATGAGATCTTTTTCTTTGTCAACAAACTTGCAAGAAAGTTTAAGAAGTAAG
GAATGA, TGTGATCTGCCTCAAACCCACAGCCTGGGTAGCAGGAGGACCTTGATGC
TCCTGGCACAGATGAGGAGAATCTCTCTTTTCTCCTGCTTGAAGGACAGACATGACT
TTGGATTTCCCCAGGAGGAGTTTGGCAACCAGTTCCAAAAGGCTGAAACCATCCCTG
TCCTCCATGAGATGATCCAGCAGATCTTCAATCTCTTCAGCACAAAGGACTCATCTG
CTGCTTGGGATGAGACCCTCCTAGACAAATTCTACACTGAACTCTACCAGCAGCTGA
ATGACCTGGAAGCCTGTGTGATACAGGGGGTGGGGGTGACAGAGACTCCCCTGATGA
AGGAGGACTCCATTCTGGCTGTGAGGAAATACTTCCAAAGAATCACTCTCTATCTGA
AAGAGAAGAAATACAGCCCTTGTGCCTGGGAGGTTGTCAGAGCAGAAATCATGAGAT
CTTTTTCTTTGTCAACAAACTTGCAAGAAAGTTTAAGAAGTAAGGAATGA and

	2	3	5	7	11	13	17	19	23
29	31	37	41	43	47	53	59	61	67
71	73	79	83	89	97	101	103	107	109
113	127	131	137	139	149	151	157	163	167
173	179	181	191	193	197	199	211	223	227
229	233	239	241	251	257	263	269	271	277
281	283	293	307	311	313	317	331	337	347
349	353	359	367	373	379	383	389	397	401
409	419	421	431	433	439	443	449	457	461
463	467	479	487	491	499	503	509	521	523
541	547	557	563	569	571	577	587	593	599
601	607	613	617	619	631	641	643	647	653
659	661	673	677	683	691	701	709	719	727
733	739	743	751	757	761	769	773	787	797
809	811	821	823	827	829	839	853	857	859
863	877	881	883	887	907	911	919	929	937
941	947	953	967	971	977	983	991	997	



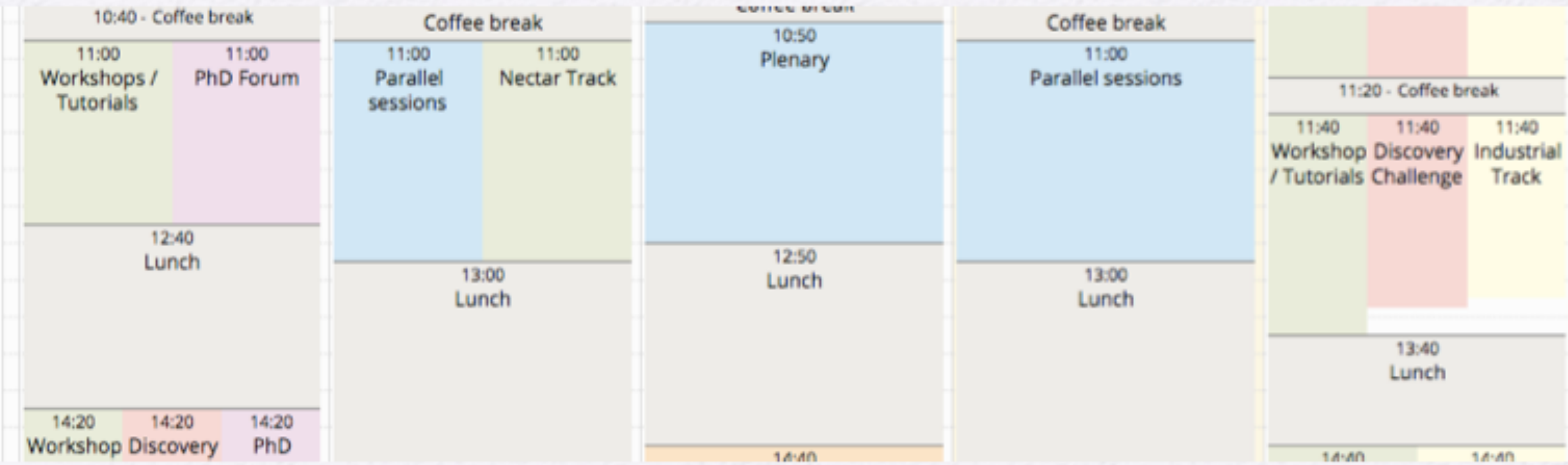
Behavioral sequences

Characteristics

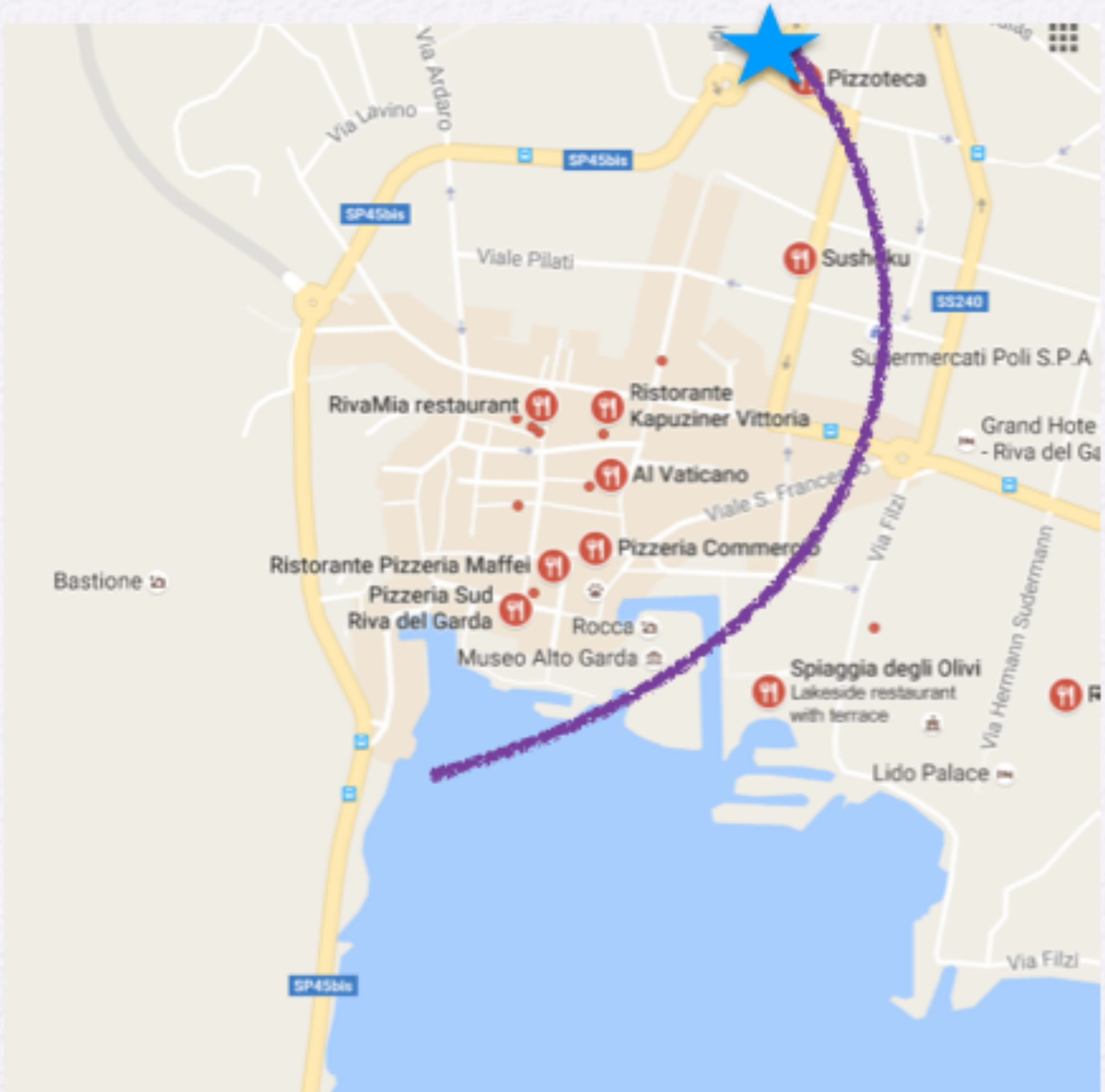
- Human generated
- Driven by structured dynamics, ie, not random
- Depends on or influenced by currently available choices, which
  - may or may not be explicit
  - might be influenced by the past
- Captures underlying choice dynamics

- Can gain insights by studying sequences
  - Properties of the items, eg, quality
  - Properties of the users, eg, propensity

Explicit choice set



Implicit choice set



Luce's axioms

- R. Duncan Luce (1925–2012), mathematical psychologist
- Independence of Irrelevant Alternatives (IIA): Probability of choosing a over b is independent of the presence of c [Luce59]
- If s_i is score of item i, then given a choice set X, probability i is chosen is: $s_i / \sum_{j \in X} s_j$
- Extendible to case when items and/or users have attributes
  - Equivalent to multinomial logistic regression

Choice: Modeling

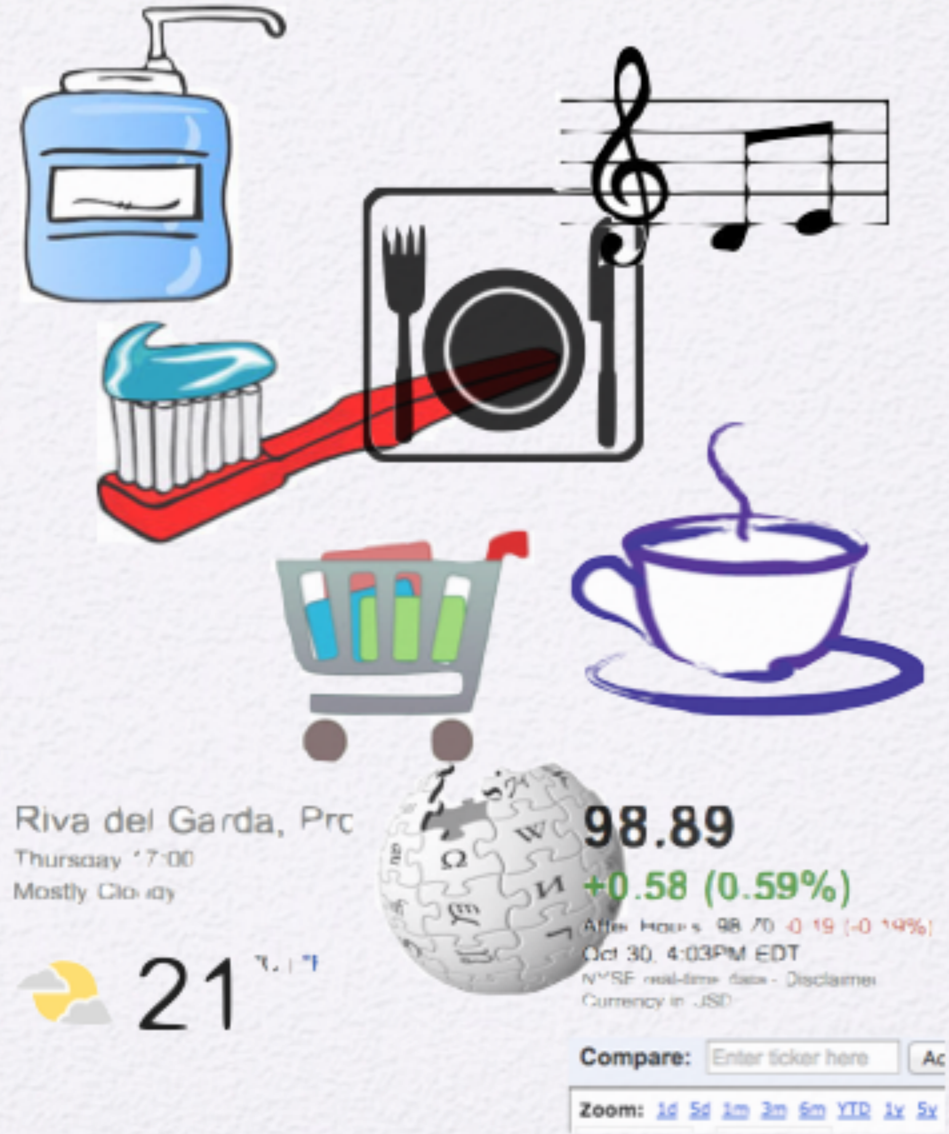
- What factors affect choice of the next item?
  - Attributes of the user and the item
  - Choice set under consideration
  - Historical preferences

- How to model this choice?
  - Luce’s axiom
  - Simple probabilistic processes
  - Markov chains/random walks

Consumption from history

Consumption from history [AndersonKTV14, BensonKT16]

- Most items users consume are not for the first time
- Users can be variety-seeking
- Sometimes go for reliability
- Sometimes go for novelty
  - Boredom
  - Try new options



Repeat consumer choice

- Marketing studies
- Consumer behavior
- Music listening experiment [KahnRK97]
  - Melioration/overconsumption: listen to favorite on each trial
  - Maximization: preserve the high level of enjoyment

- Possible explanations
  - Difficulties in prediction of taste
  - Users try to create the best memory (five flavors vs one flavor)
  - Zen principles (pain vs pleasure)

Re-searching

- Repeat queries in search logs [Teevan et al]
- 40% of queries are re-finding queries
- Navigational queries are more likely to be repeated
  - Information re-finding

- Repeat behavior leads to easier prediction of which results will be clicked

Re-visiting web pages

Re-visiting web pages

- Web page revisitation using browser logs [Adar et al]
- 50–80% of the web pages are revisited
- Revisitation reasons
  - Bookmarks/use as hub
  - Track content change
  - Backbutton
- Types of revisitation
  - Fast: shopping pages, references, traffic
  - Medium: mail, forums, news, ...
  - Slow: weekend activity, software updates, ...

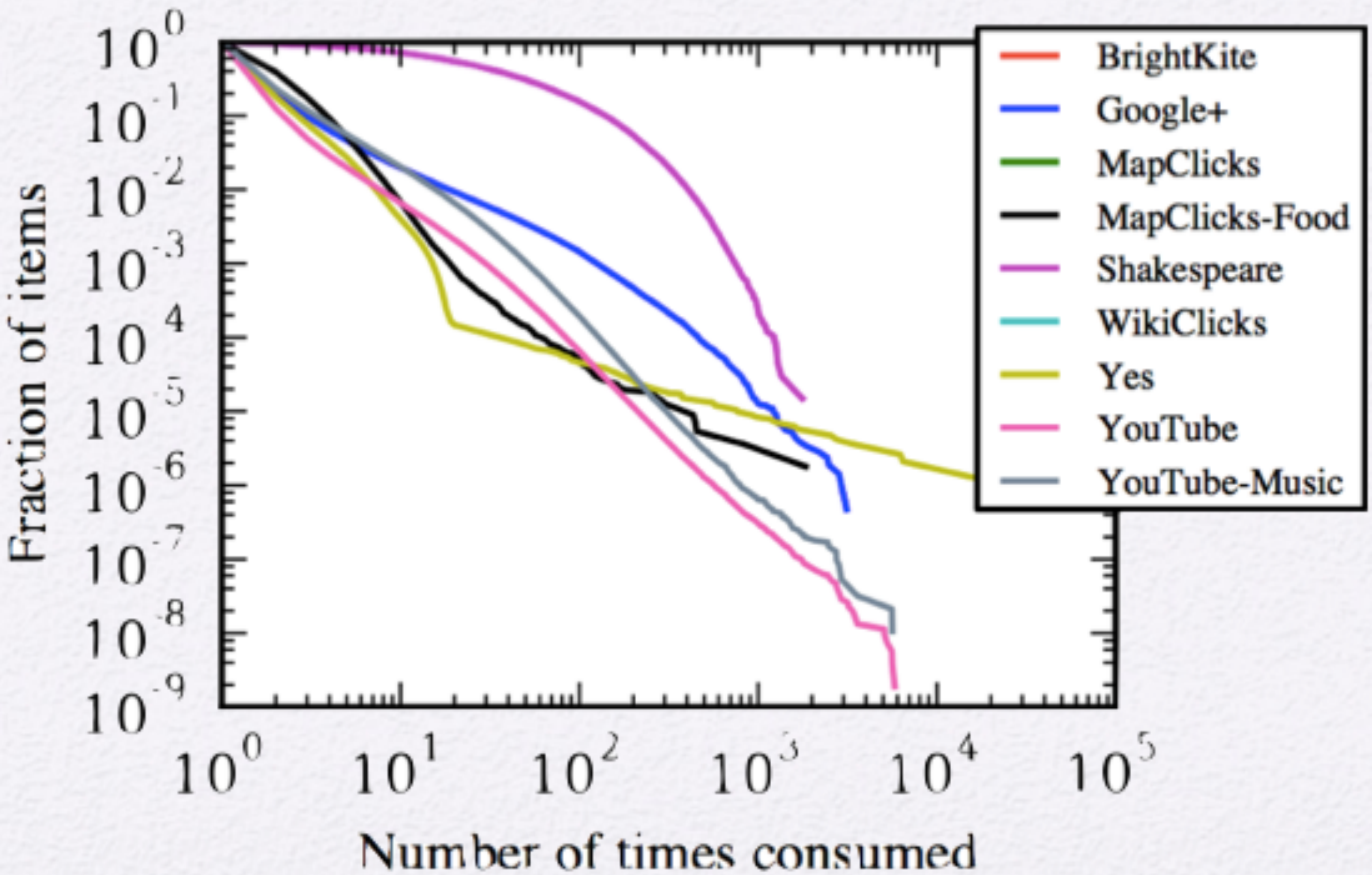
Domains of reconsumption

Domains of reconsumption

- Location checkins
  - BrightKite
  - Google+
- Clicks
  - Businesses on maps
  - Restaurants on maps
  - Wikipedia
- Media
  - Youtube,music videos
  - Music songs, artists
  - Playlists from radio stations
- Shakespeare! 

Does it exist?

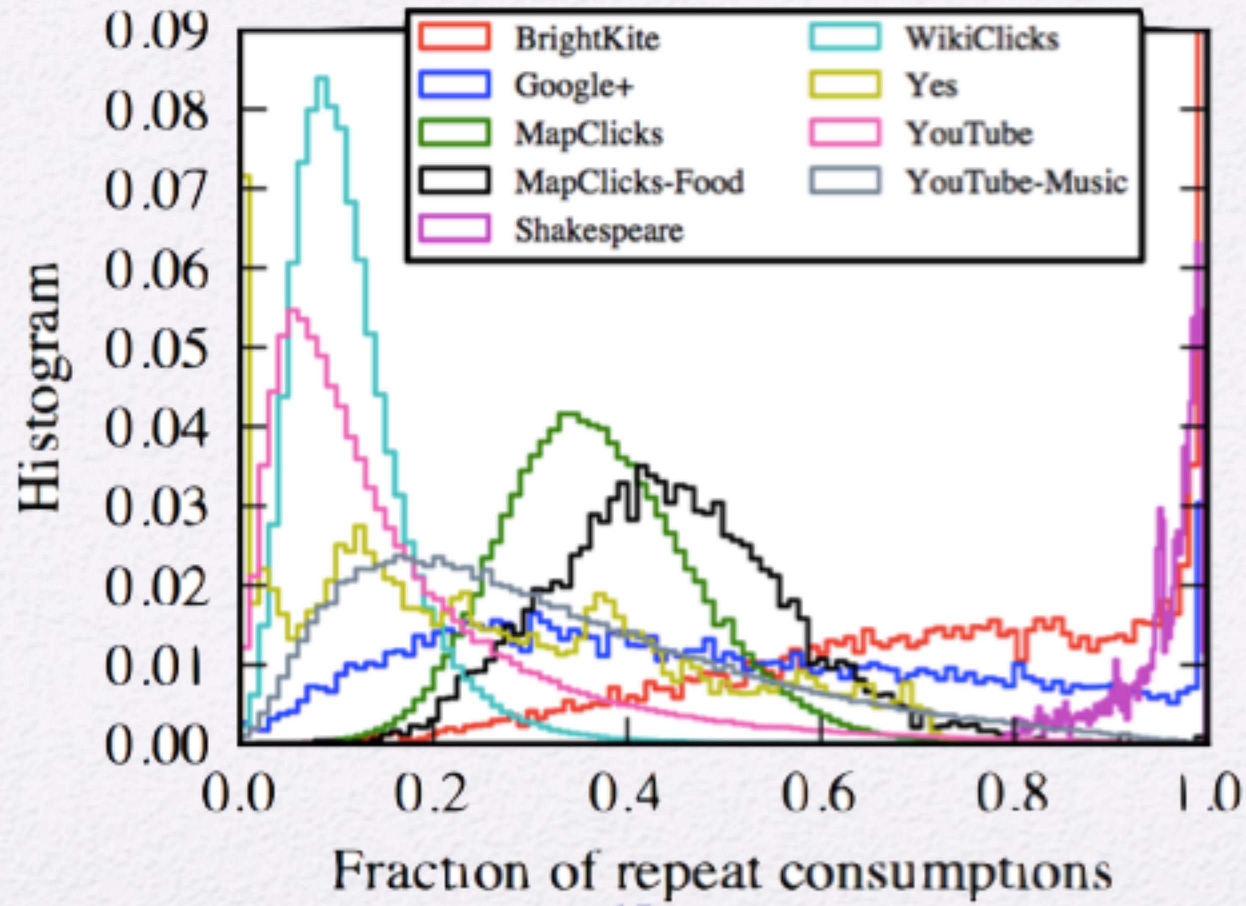
Distribution of the #times an item is consumed



Does it exist?

Does it exist?

Distribution of the fraction of repeat consumption (over each user)



Reconsumption properties

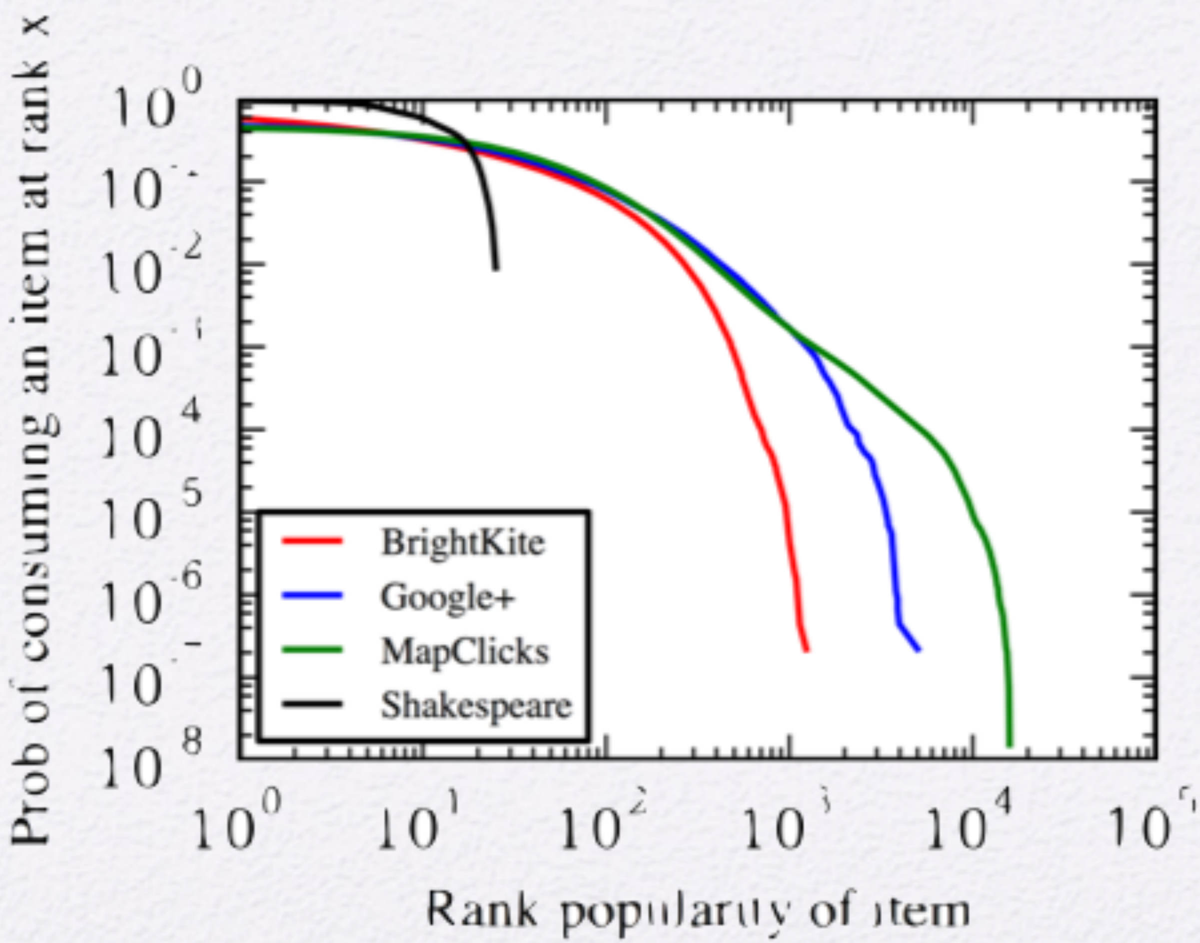
Reconsumption properties

- What are the empirical traits of reconsumption and reconsumed items?
- Are users exploring or exploiting?
- What is the role of item popularity?
- What are the roles of time and recency?

Popularity vs reconsumption

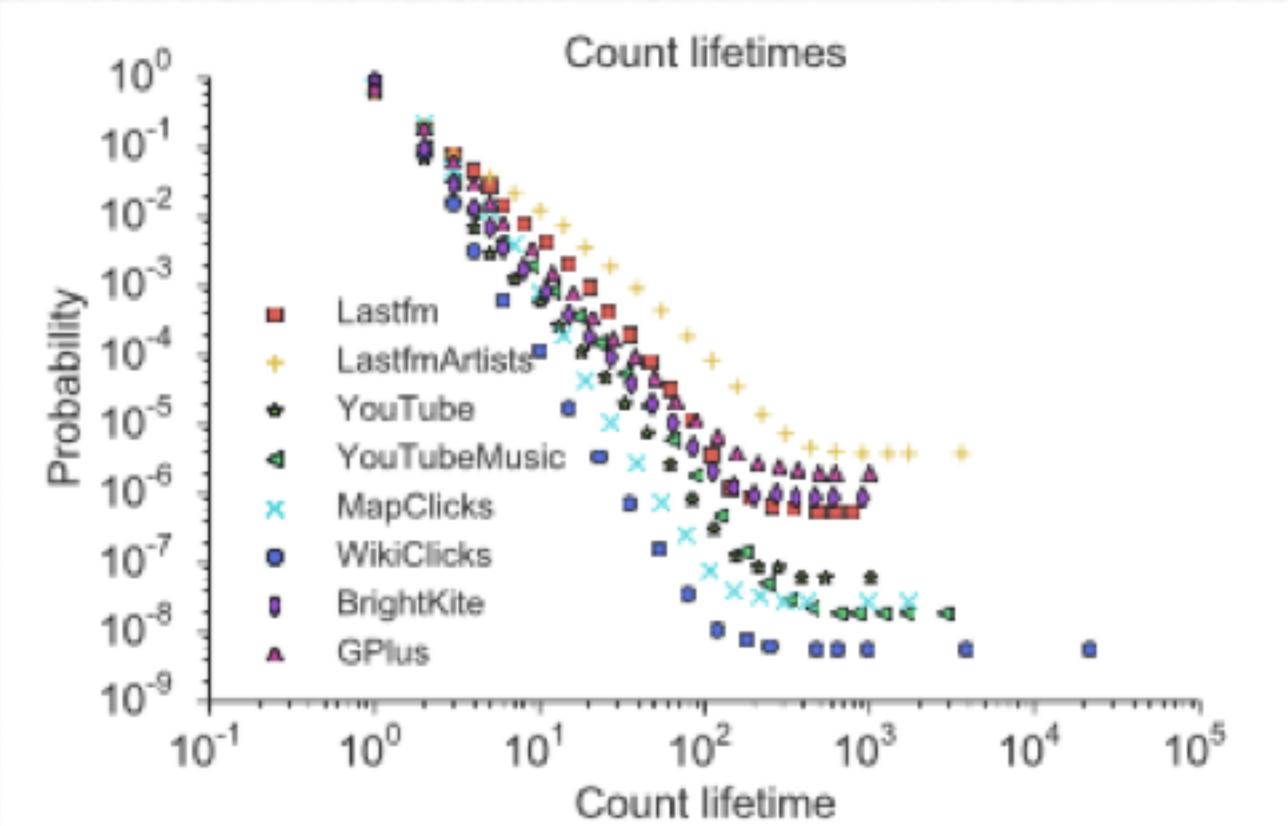
Popularity vs reconsumption

More frequently consumed items are more likely to be reconsumed (low rank means more popular)



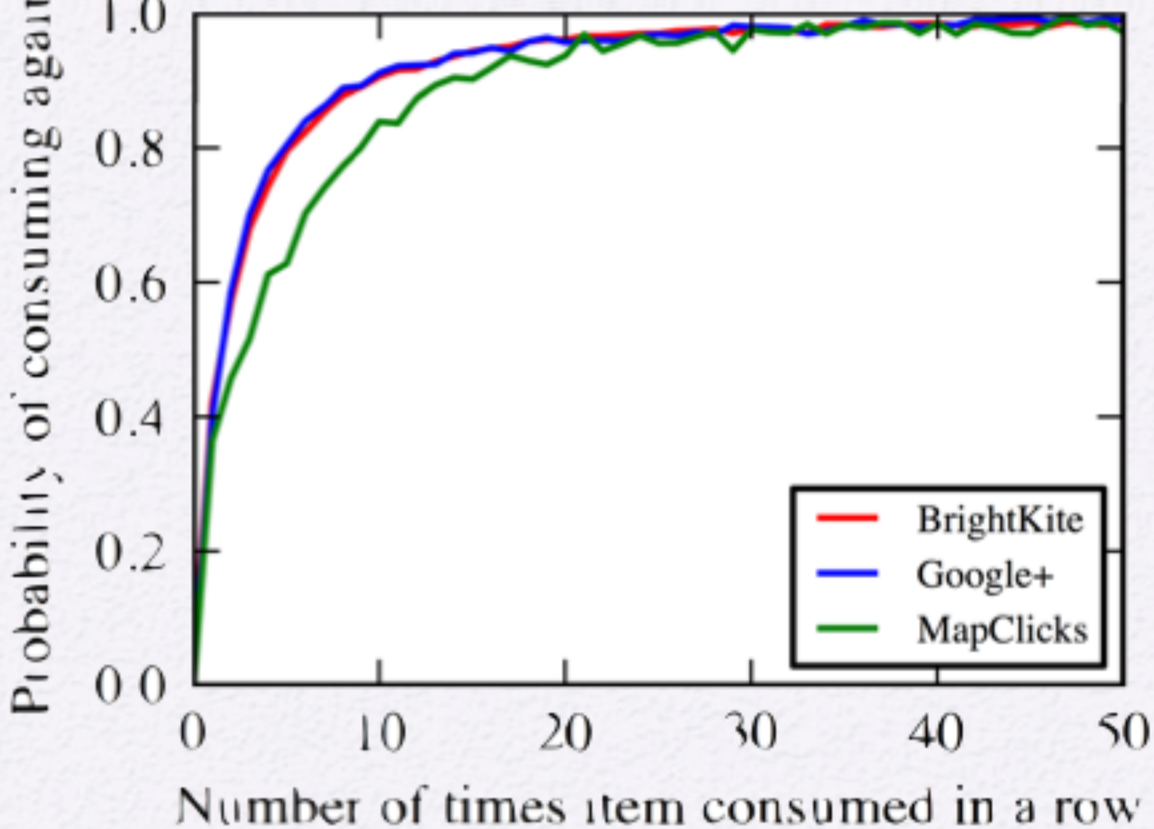
Lifetime distributions

Do items have finite lifetimes?



Of the items in the first 20% of consumption sequences, >85% do not appear in the final 20% of the sequence

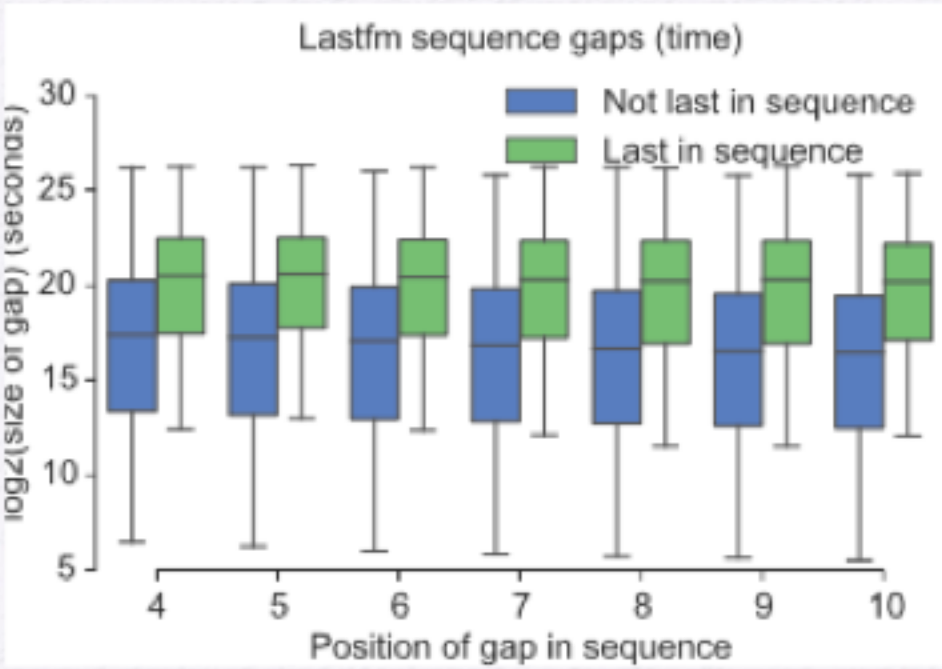
Satiation

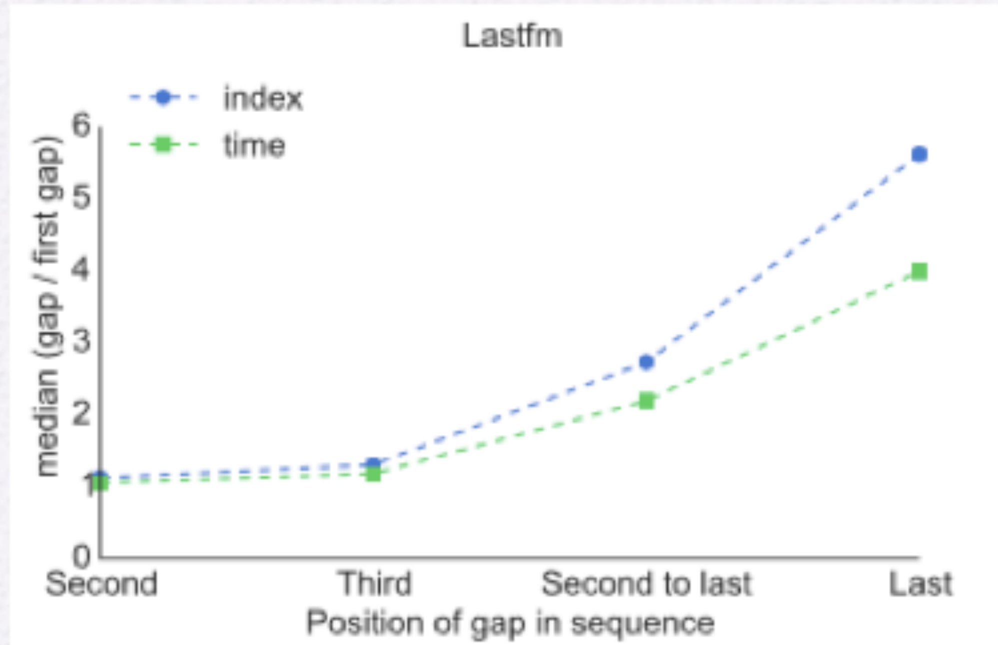


No evidence of satiation in online user behavior

Boredom

- Do users get bored with repeat consumption?
  - Marketers, advertisers care about this
  - Churn/variety-seeking behavior





As a user begins to abandon an item, the gap between repeat consumptions of that item grows

Studying recency effects

- How does the recency of consumption affect the likelihood of reconsumption?
  - Is recency really a factor?
- Study this using ideas from analyzing performance of caches
  - Hit-rate curves

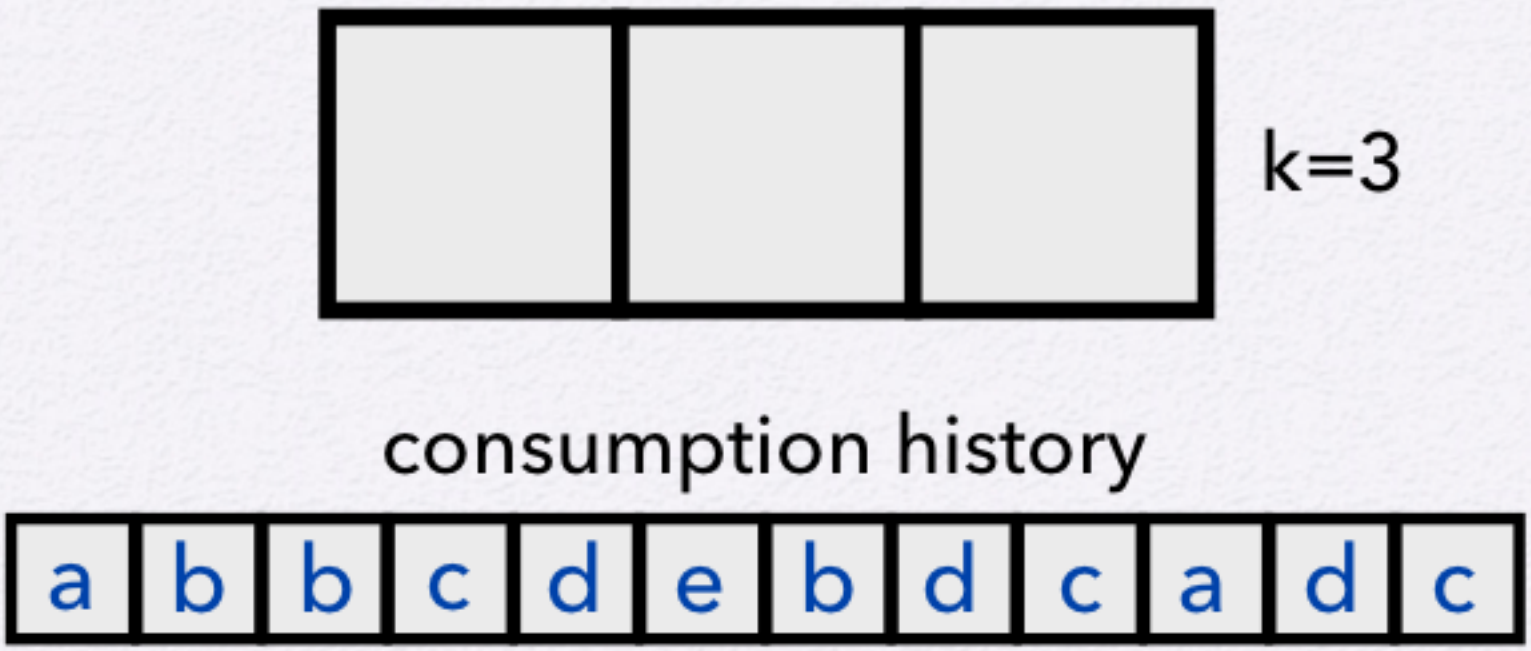
Consumption with a cache

An example

Fix a cache size k

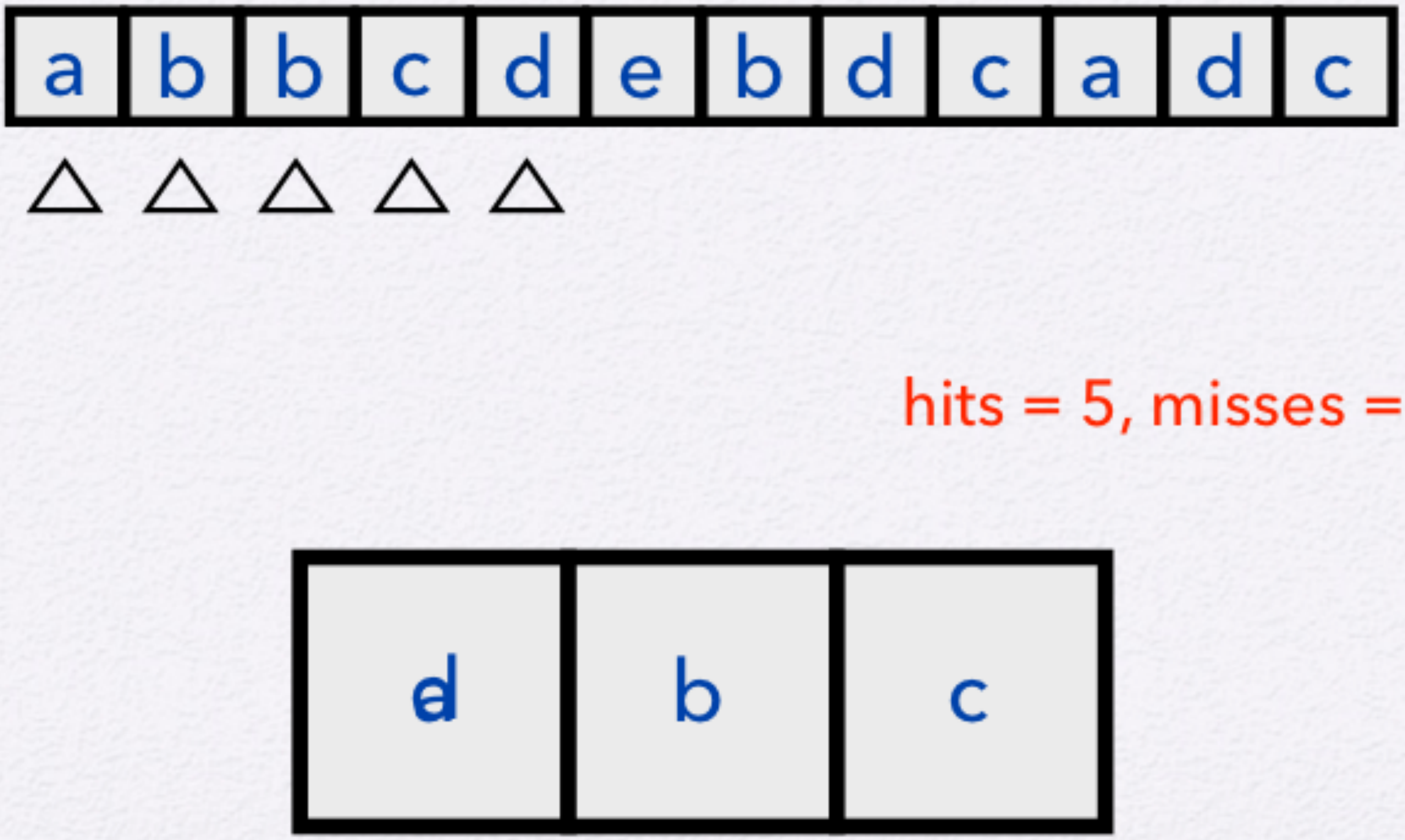
Process consumption history using an optimal offline caching algorithm

- Replace item that occurs furthest in the future



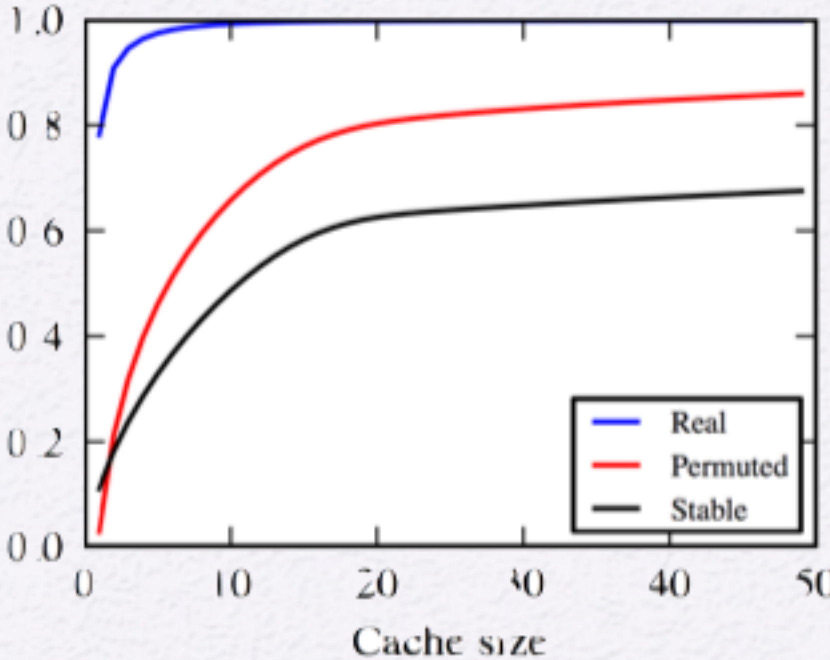
Normalized hit-rate curves

consumption history

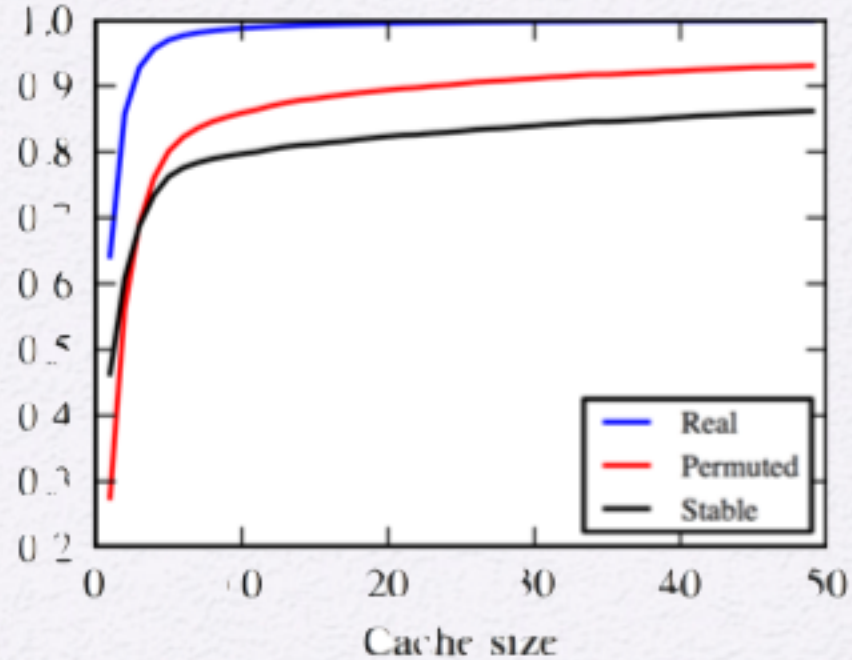


hits = 5, misses = 7

- Hit-ratio is the degree to which recency is observed in consumption history
- Compute for various cache sizes

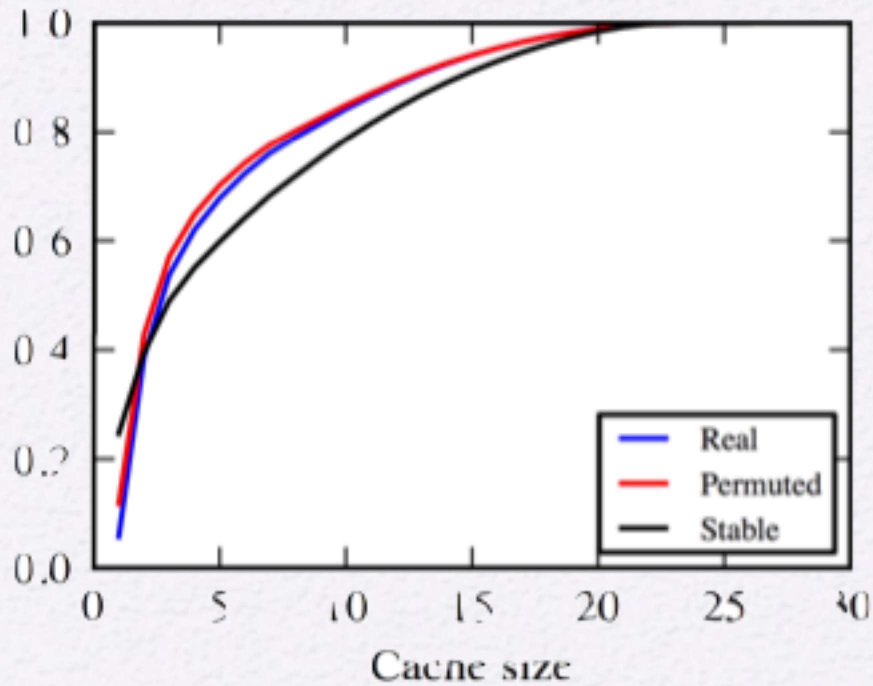


(i) MAPCLICKS

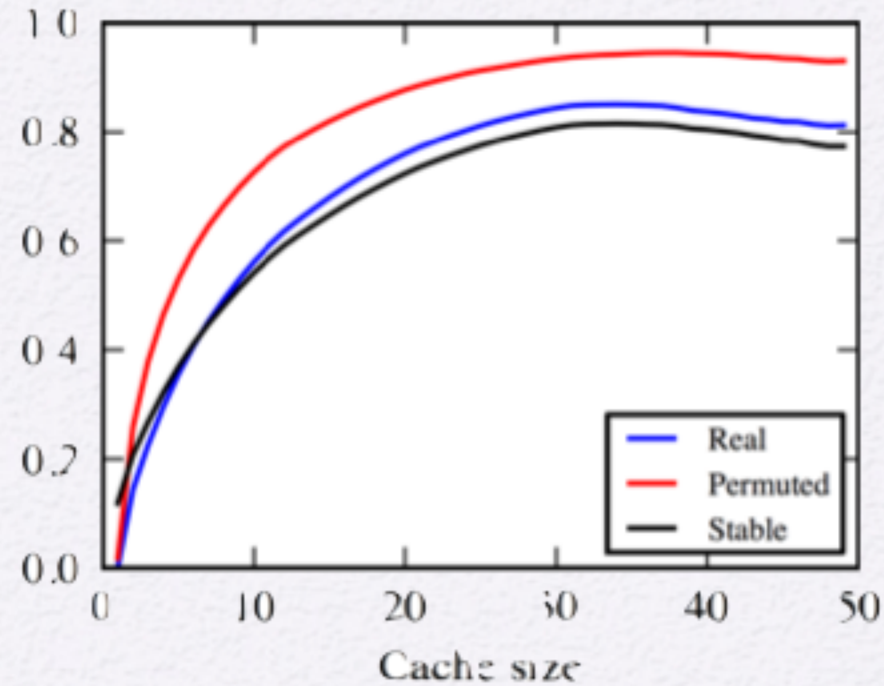


(ii) BRIGHTKITE

No recency effects



(iii) SHAKESPEARE



(iv) YES

Three key factors

Factors

- How popular is the item?
- Time gap since it was last consumed
- #other items consumed more recently

Can we develop a holistic mathematical framework powerful yet simple enough to explain patterns of reconsumption we observe in real data?

Formulation

- Fix a vocabulary of items
  - Can be videos, albums, songs, restaurants, ...
- Consumption history for a user is a time-ordered sequence of items x_1, x_2, ..., each from the vocabulary

Model: which item would the user picks to consume next?

A holistic model

Three stages:

Three stages:

- Temporal step
  - Model inter-arrival times between items
- Novelty step
  - Which items are novel, which items are repeat?
- Repeat choice step
  - How to choose the repeat items?

No planted notions

Temporal model

Goal: model the times of consumption

Goal: model the times of consumption

- Simplest: iid choices for inter–arrival times
  - Fails to capture bursty behavior

- A semi–Markov model: intra– and inter–session behaviors separated
  - Number of items in a session (Power law)
  - Intra–session gap (Double Pareto)
  - Inter–session gap (Power law)

Novelty model

Predict the probability of repeat consumption

Predict the probability of repeat consumption

- Use logistic regression with features
  - Proclivity for reconsumption
  - Number of items consumed so far
  - Time since last reconsumption
  - Were the last few times novel?

- This gets accuracy of > 0.8 on most datasets
- Predicting novelty is a well-studied problem
  - Recommender systems

Recency model

- Empirically, recency seems to play a strong

- Empirically, recency seems to play a strong role in reconsumption
- Formulate a copying model [Simon55] based on recency

- Each gap in position p has a weight w(p)
- At each step i, user picks item e at position j with probability proportional to w(i–j)
  Ie, at position i, pick item e with probability

$$\frac{\sum_{j<i} [x_j = e]\, w(i-j)}{\sum_{j<i} w(i-j)}$$

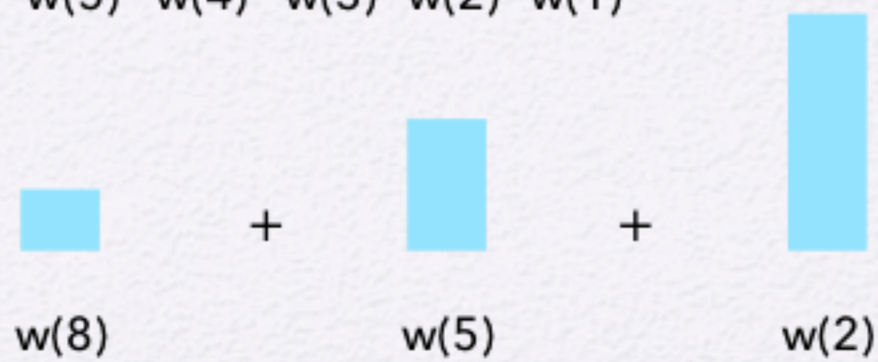
Example

consumption history

a	b	b	c	d	e	b	d	a	c	d	c	?

weights *w*

Pr[d is consumed next] ~



Score-based model

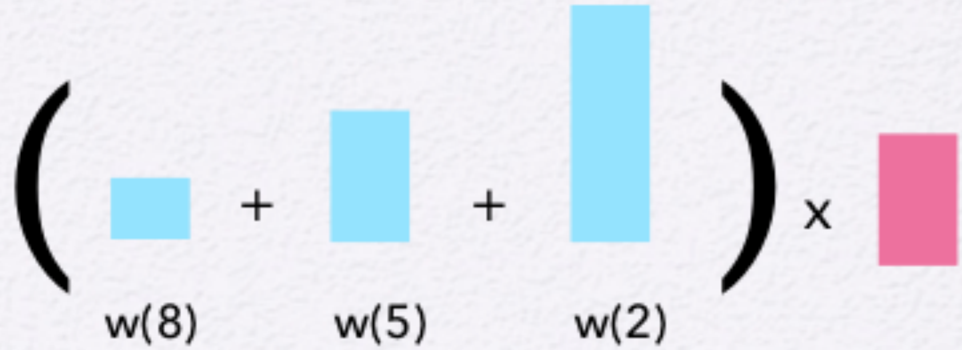
- Each item e has a score s(e)

- The score reflects the quality of the item
- The score dictates the reconsumption pattern
- Pick next item e with probability

$$s(e) \,/\, \sum s(e')$$

Recency + quality model

- Combine recency and quality

Pr[d consumed next] ~ 

At position i, pick item e with probability

$$\frac{\sum_{j<i} [x_j = e]\, w(i-j)\, s(x_j)}{\sum_{j<i} w(i-j)\, s(x_j)}$$

The full model

- i–th item consumed
  at time t

at time t_i

- Combine recency, quality, and time

$$p(i - j, x_j, t_i - t_j)$$

#items consumed since j

quality of x_j

elapsed time

At position i, pick item e with probability

$$\frac{\sum_{j<i} [x_j = e]\, w(i-j)\, s(x_j)\, T(t_i - t_j)}{\sum_{j<i} w(i-j)\, s(x_j)\, T(t_i - t_j)}$$

Learning the choice models

Simplest case

Score model: score of an item is given by the empirical fraction of its occurrences

$$s(e) = (1/k) \sum_{i<k} [x_i = e]$$

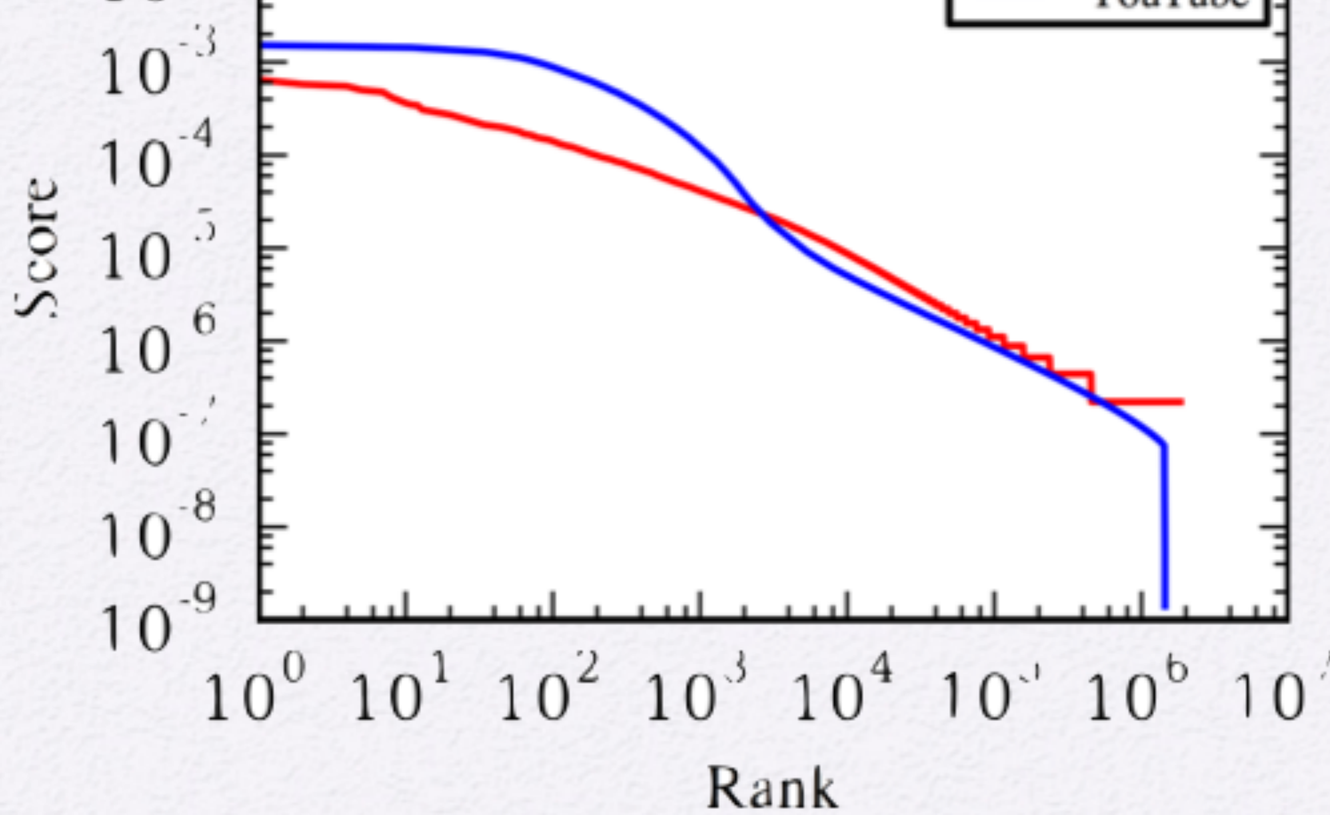
Learning the full model

- Maximize likelihood

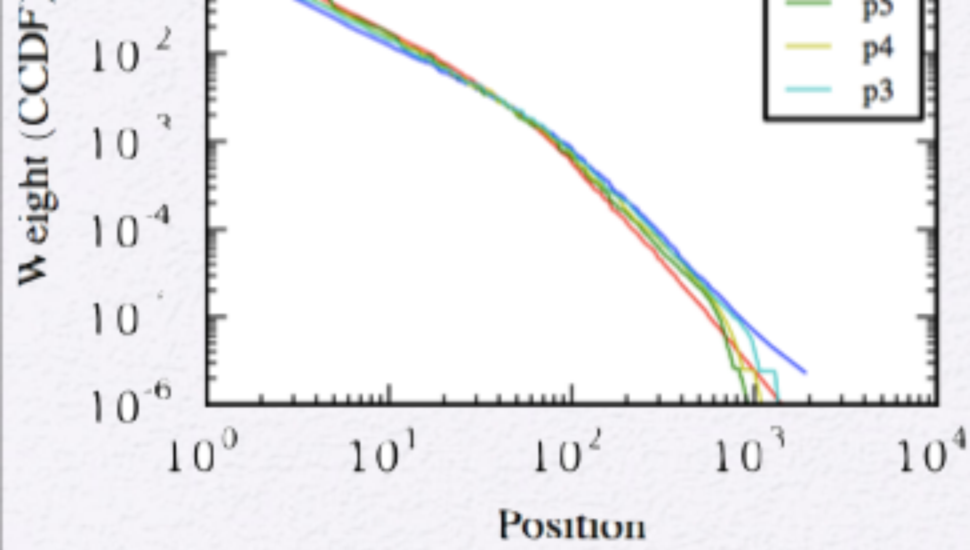
$$\log \prod \frac{\sum_{j<i} [x_j = x_i]\, w(i-j)\, s(x_j)\, T(t_i - t_j)}{\sum_{j<i} w(i-j)\, s(x_j)\, T(t_i - t_j)}$$

- Stochastic gradient ascent
- Alternating updates to scores, weights, times
- Learn non-parametrically
- Not a convex problem

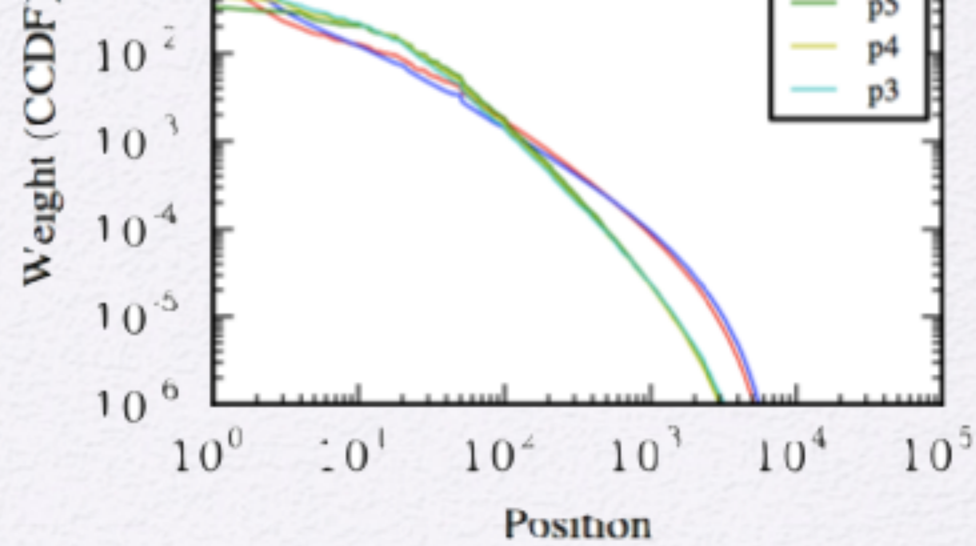
Learned quality scores



Learned recency weights

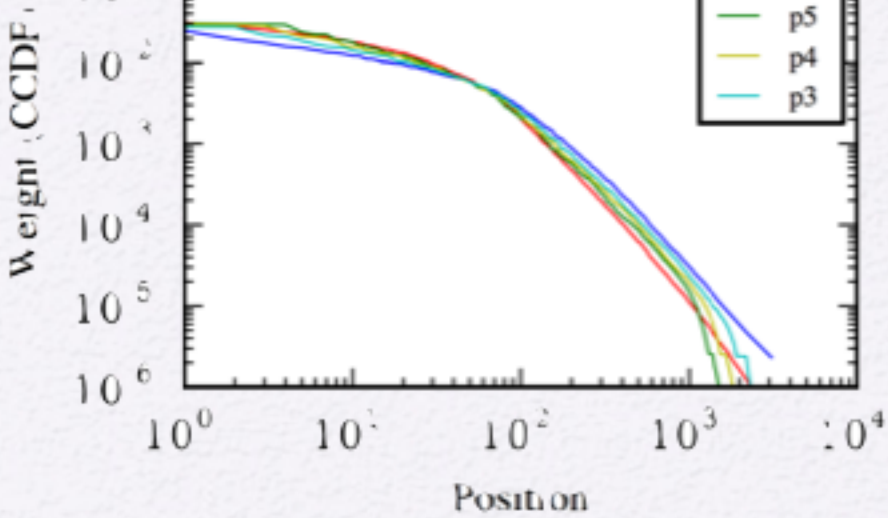


(a) GPLUS

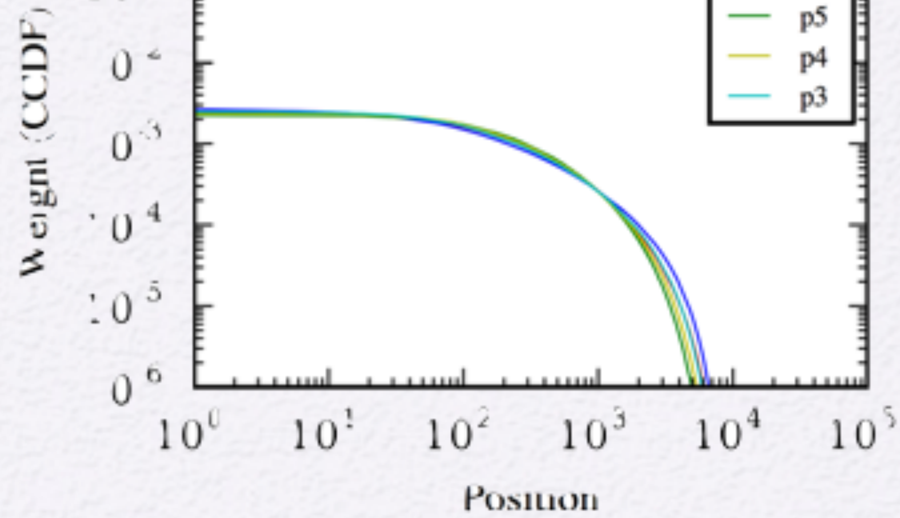


(b) YOUTUBE

Recency weights: shuffled data



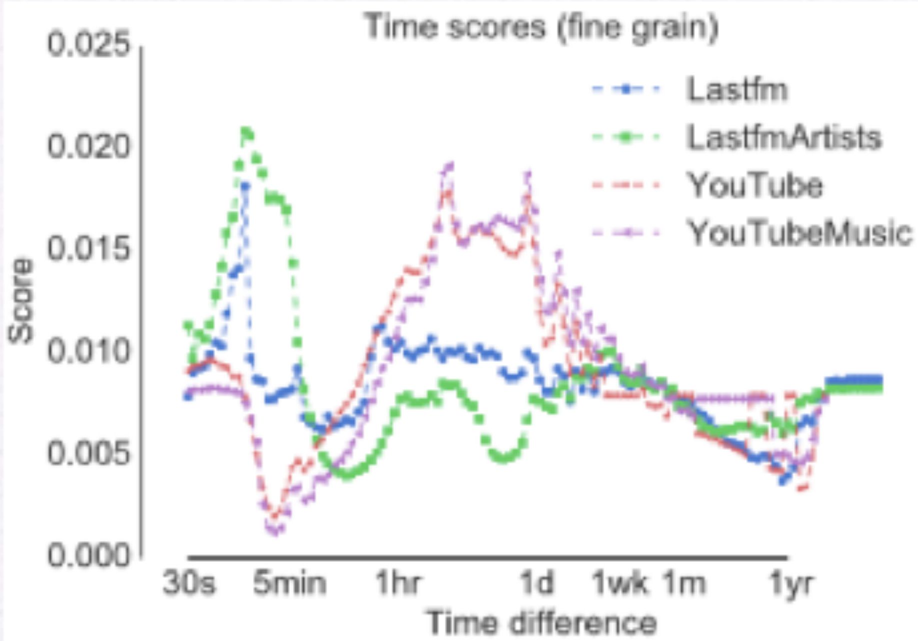
(a) GPLUS

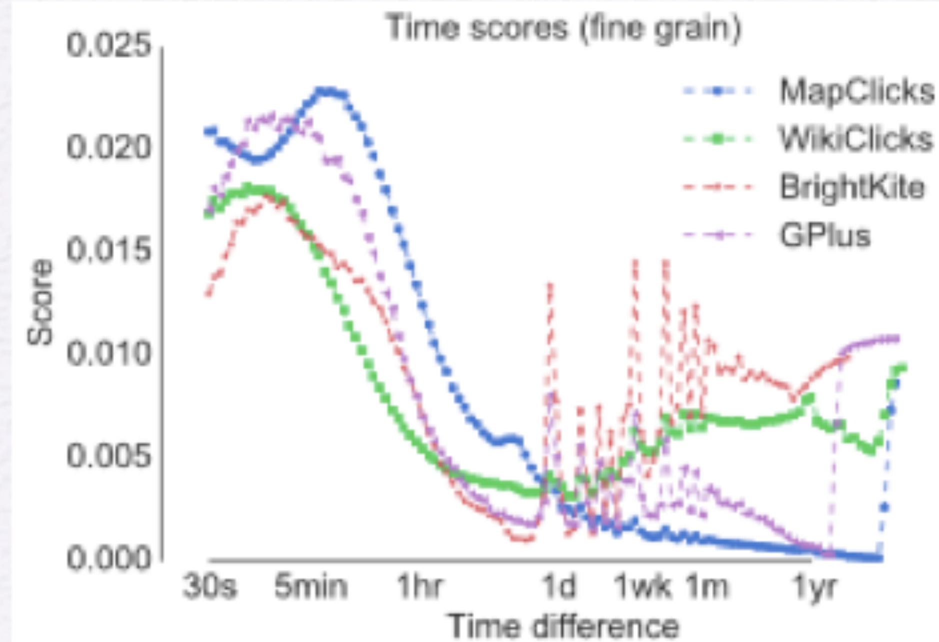


(b) YOUTUBE

Time weights

Interesting phenomenon captured by time weights (in $30*1.1^k$ second buckets)





Time weights capture, eg, cyclic behavior in checkins

Likelihoods (wrt full model)

Dataset	w	w and s	w and T
BrightKite	0.91	0.92	0.98
Plus	0.87	0.92	0.94

Plus	0.87	0.92	0.94
LastFM	0.99	0.99	1.00
YouTube	0.91	0.94	0.96
MapClicks	0.81	0.82	0.99
WikiClicks	0.78	0.81	0.91

- Recency comes close to full model
- Elapsed time plays a bigger role than quality
- Other findings:
  - Recency much better than quality
  - Popularity seems to bring the models down even with recency

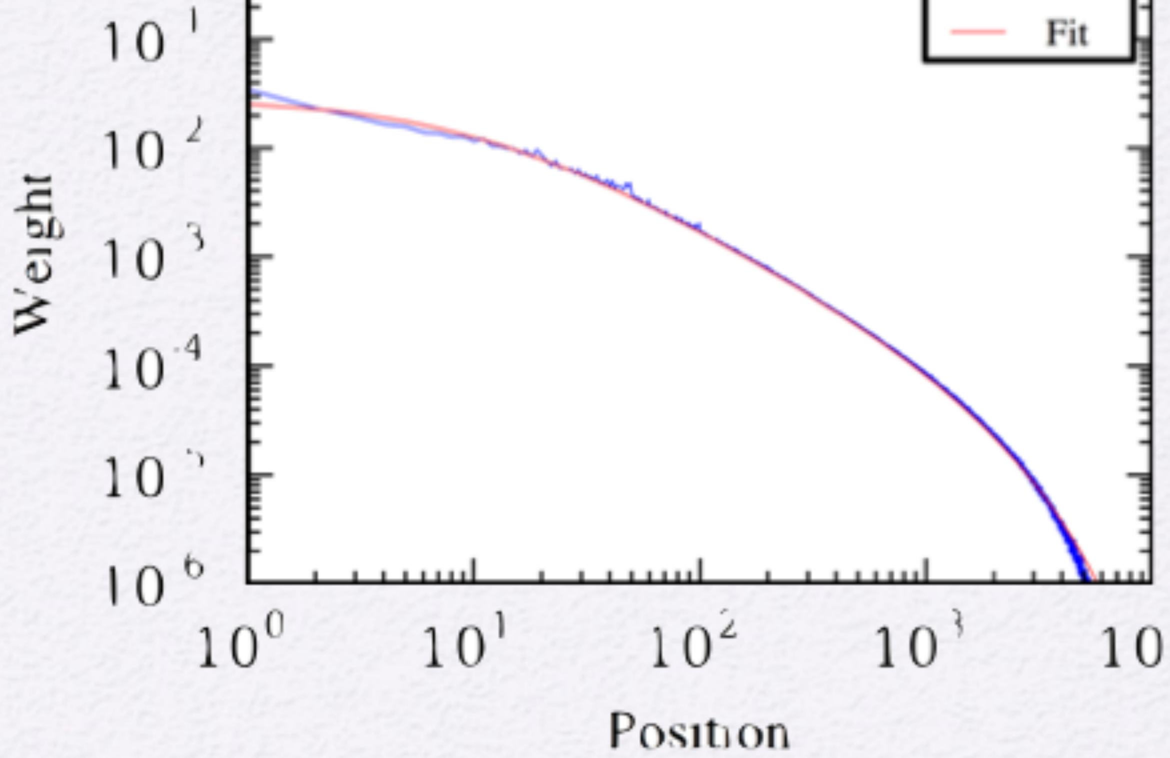
Parsimonious model

- Weights can be compressed

- Weights follow power law with exponential cutoff:

Pr[x] prop. to

$(x+c)^{-a} e^{-bx}$



Parsimonious model

Dataset	Recency@50	PLECO
BrightKite	0.654	0.926

BrightKite	0.654	0.926
GPlus	0.710	0.987
MapClicks	0.668	0.921
WikiClicks	0.971	0.999
YouTube	0.917	0.997

Recency model can be expressed using just three parameters!

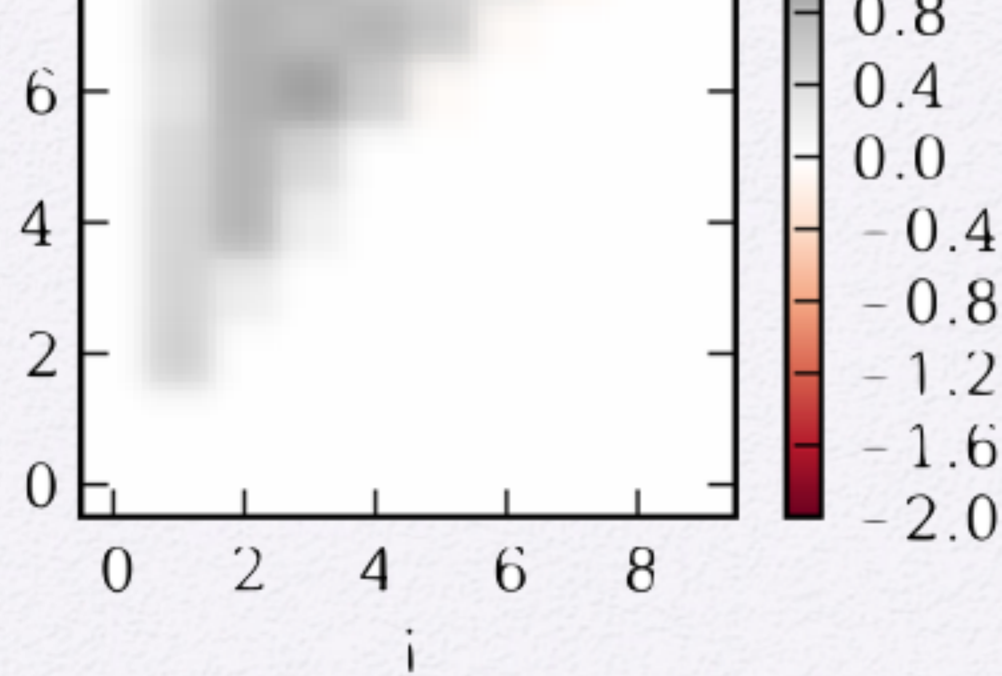
Additivity assumption

w(i)+w(j) vs w(i, j)

Getting addicted:

Getting addicted:
superadditive?

Getting bored:
subadditive?



Very small deviations from additive behavior
Mildly superadditive as popular items chosen

Personalization

Create a model for each heavy user

Dataset	Global	D. Pareto	PRLECO

Dataset	Global	D-Pareto	PPLECO
LastFM	0.80	0.89	0.88
YouTube	0.68	0.77	0.73
BrightKite	0.45	0.99	0.90

Tipping behavior

In the recency model, tipping occurs if after a certain time, only one item is repeatedly consumed

Assume weights are decreasing: $w(p) \geqq w(p+1)$

Claim. If sum of weights is finite, then tipping occurs with constant probability

Claim. If the sum of weights is infinite, then tipping does not occur

Lifetimes and boredom

Claim. If sum of weights is less than $1/p$ where p is the probability of novel item, then the lifetime is finite almost surely

- A branching process analysis

- A branching process analysis

Claim. Conditioning on the fact that the item will not be consumed again, its last gap will be larger

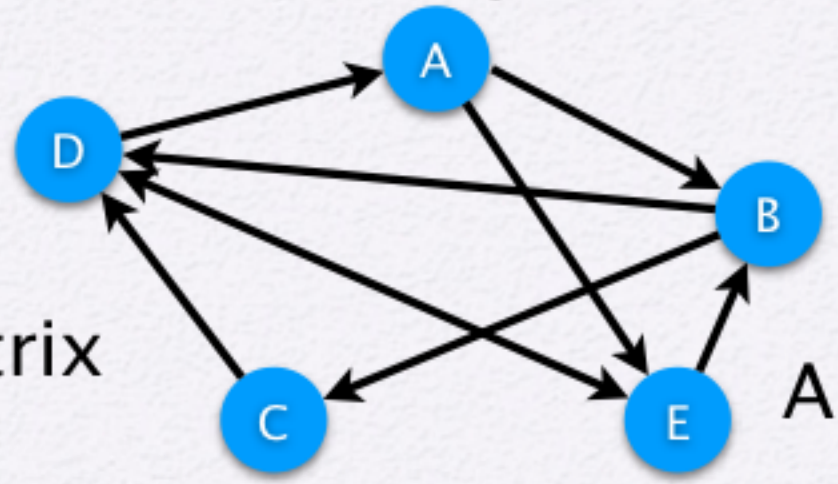
- Decreasing weights => decreasing gaps
- Captures boredom
- No need to explicitly model it

Markov chains, random walks

Markov assumption: next state depends only on current state and not on history

A simple way of modeling sequences

A simple way of modeling sequences



Transition matrix

A E B C D A B C ...

Under simple niceness conditions, there is a unique stationary distribution

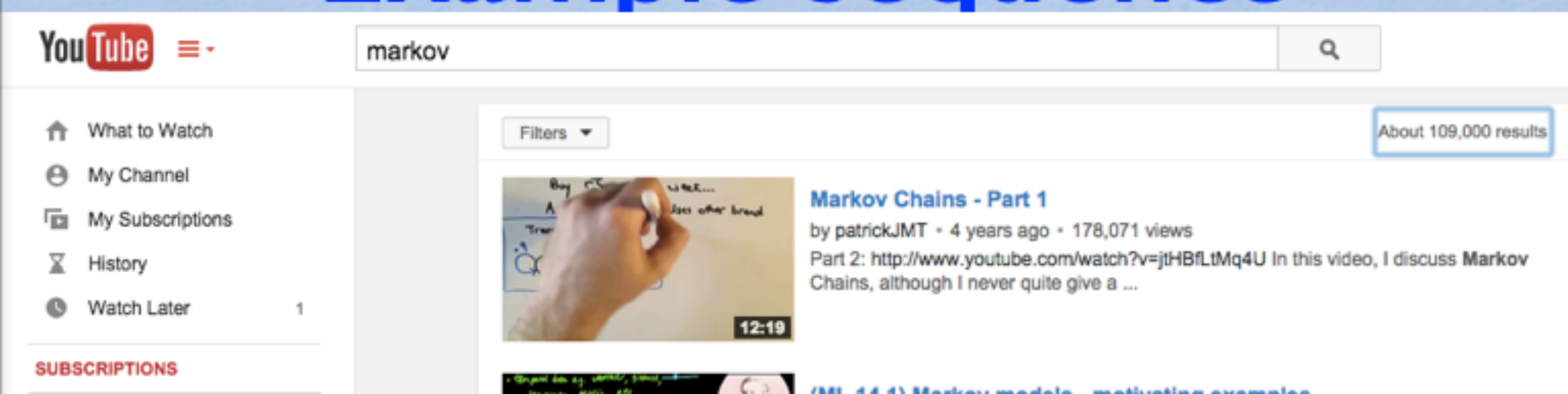
Wide use

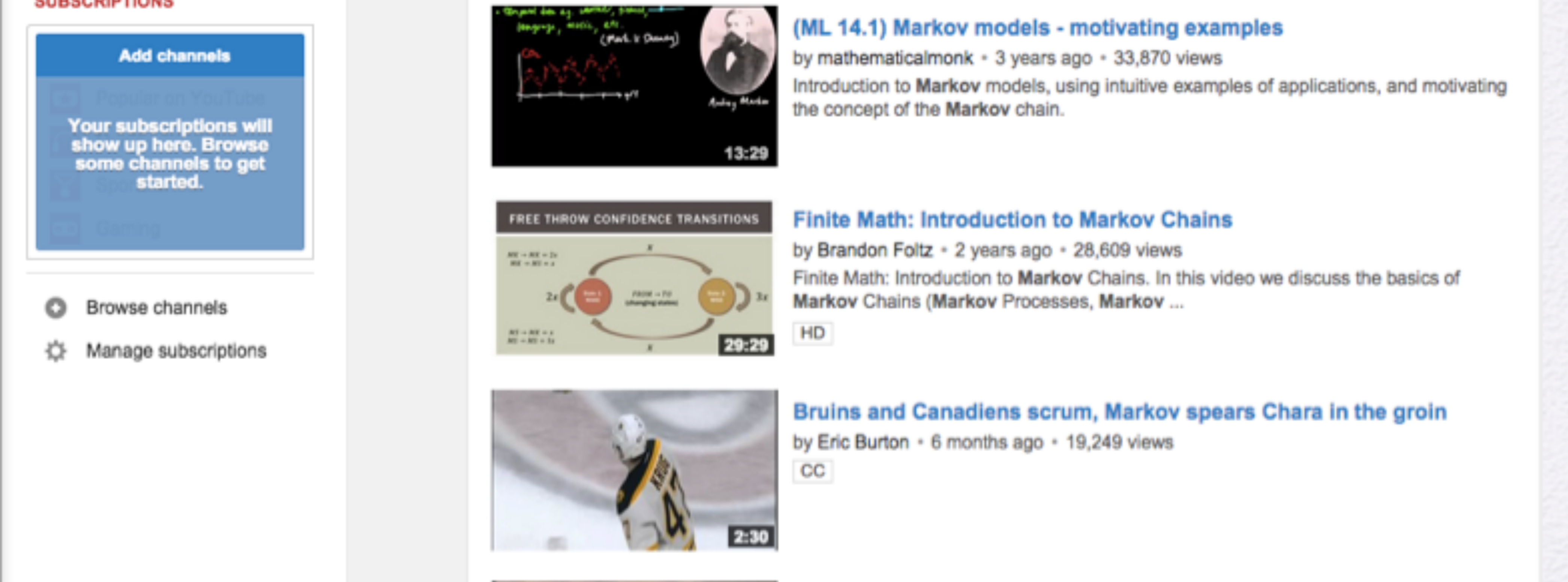
- Captures a lot of interactions
  - Typically: compute and use the stationary distribution
  - Beautiful and rich theory with great applications

• Beautiful and rich theory with great applications

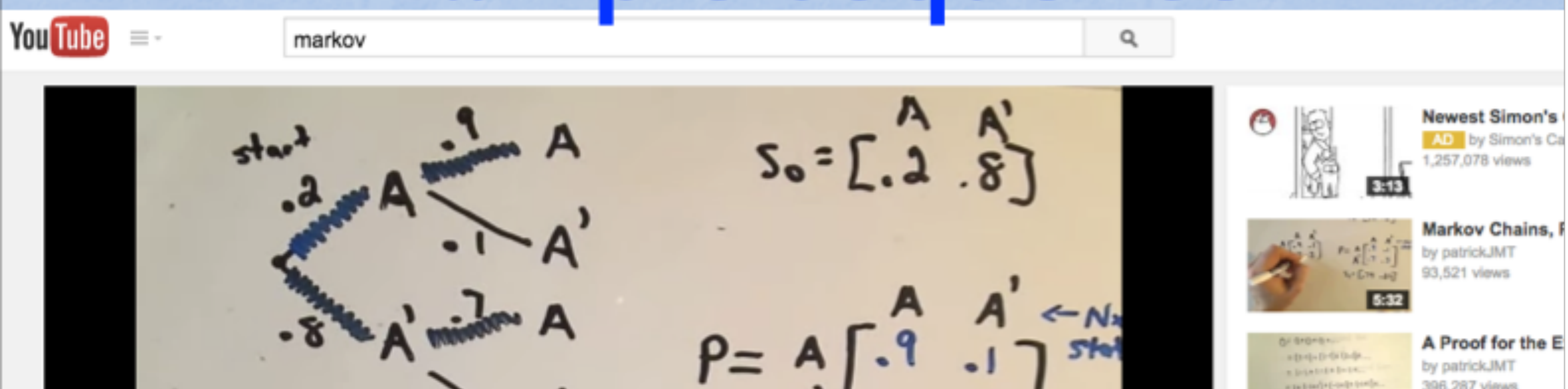
- Examples
  - PageRank: Random surfer stationary distribution
  - Translation: Use language models to build phrases

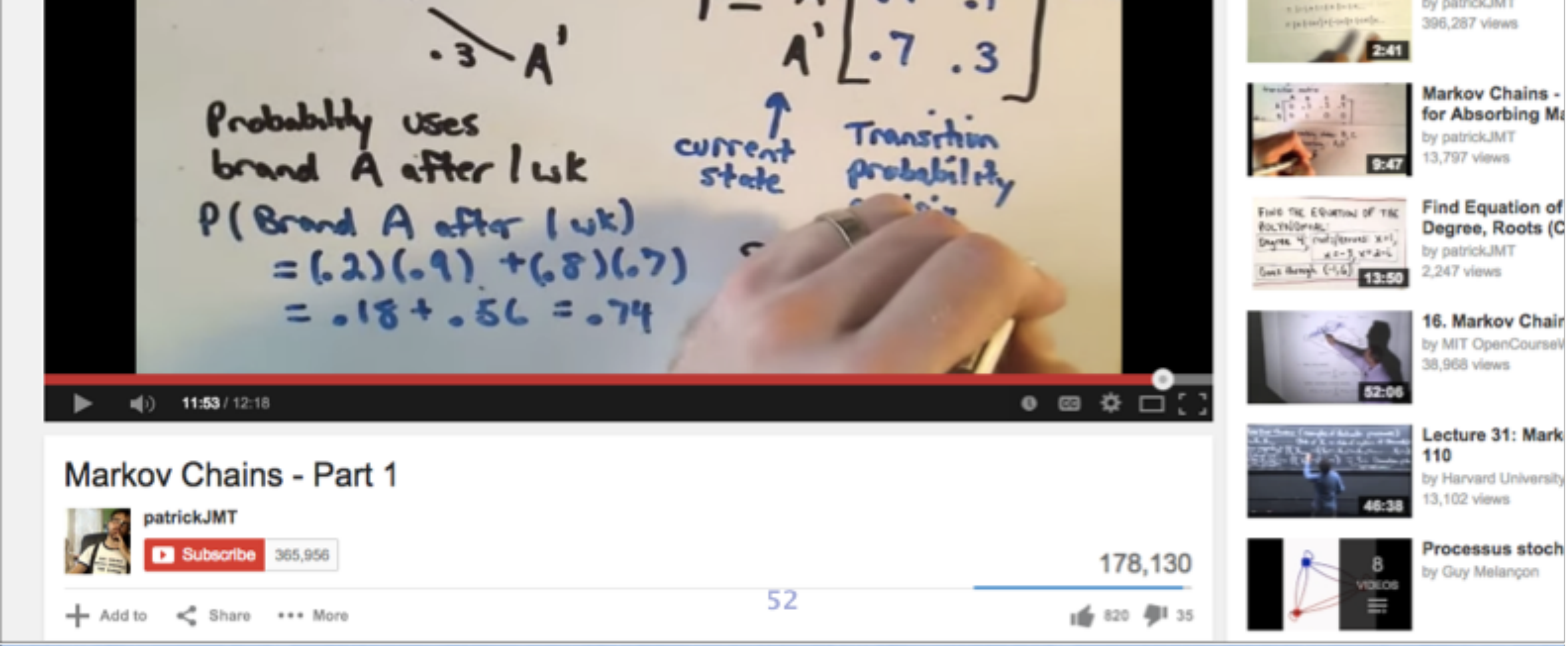
Example sequence



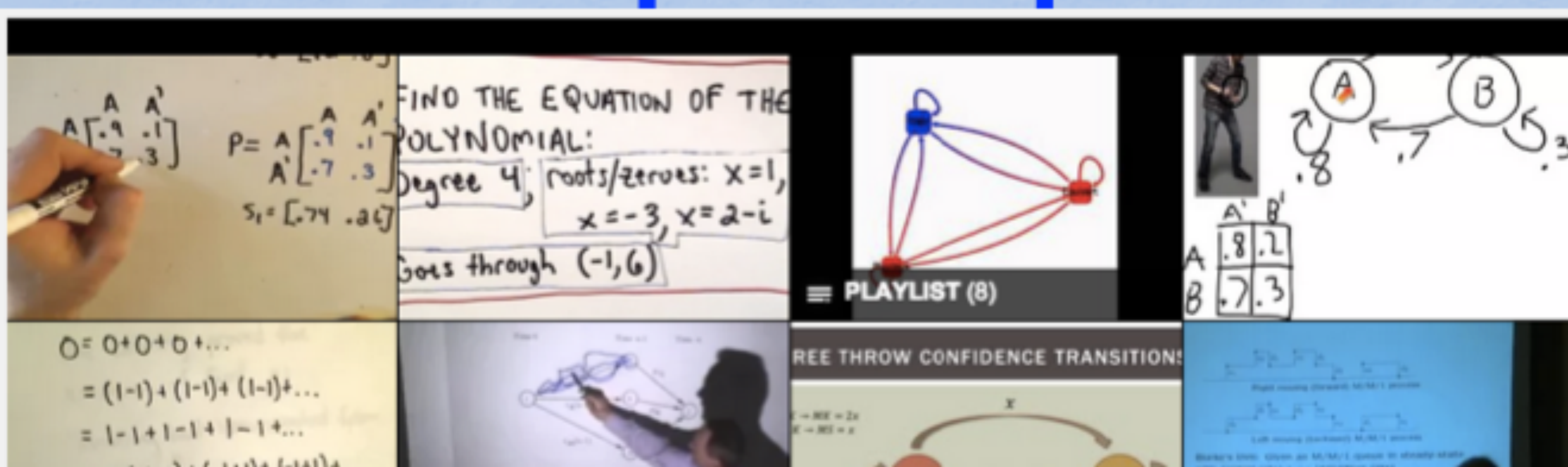


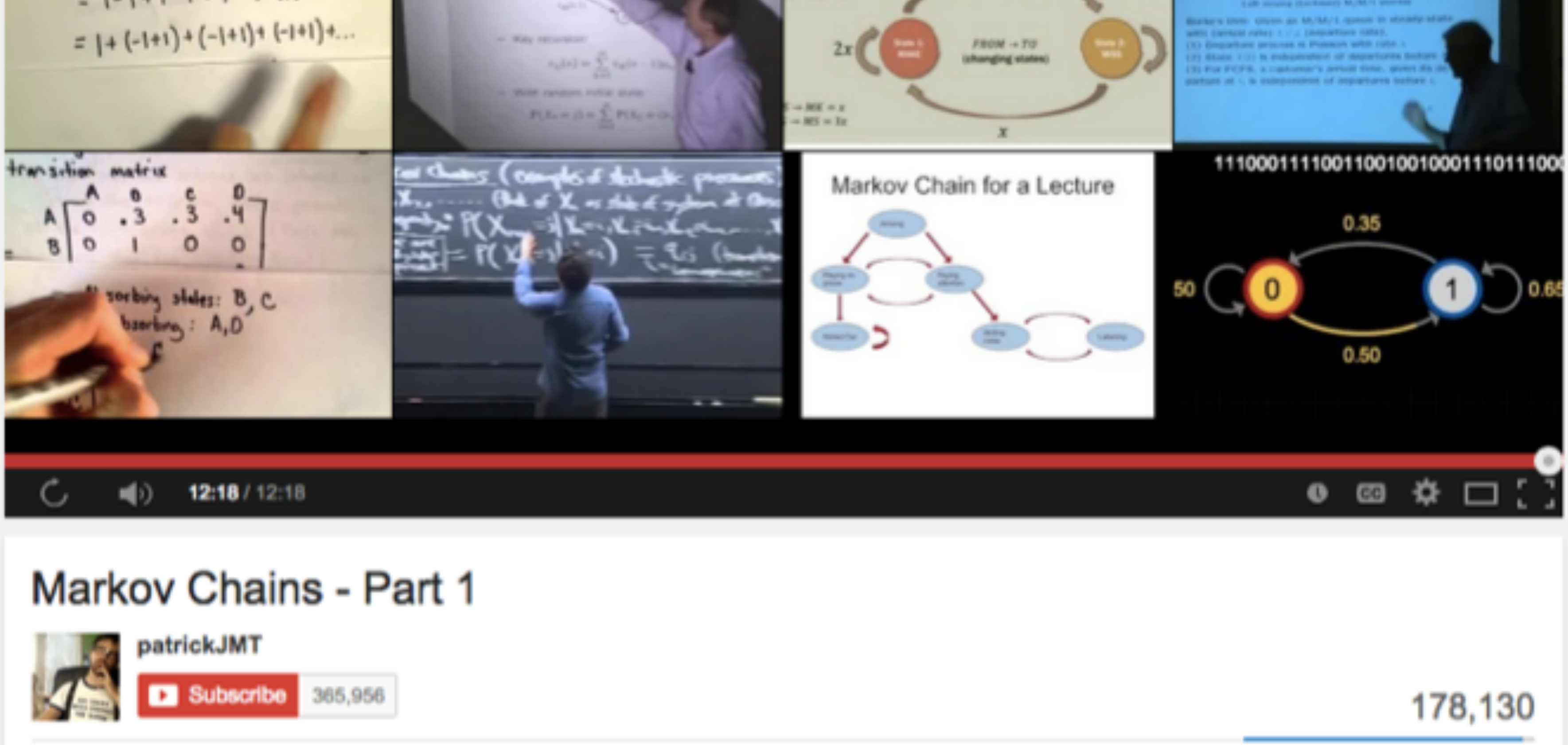
Example sequence



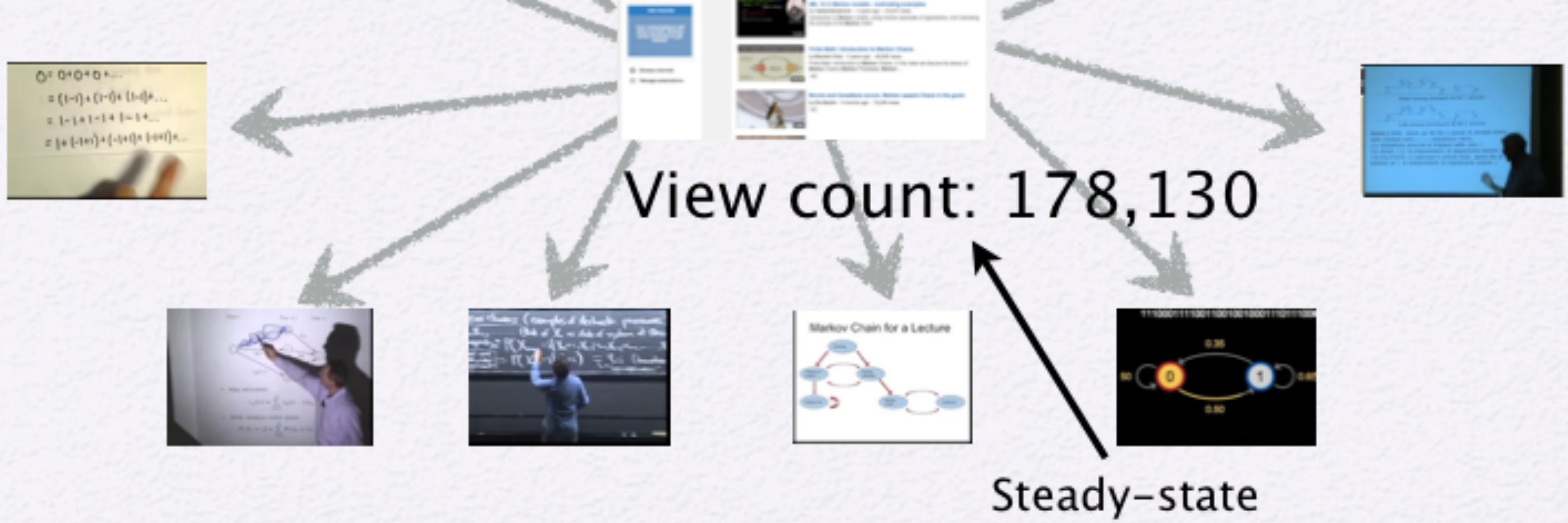


Example sequence





Example sequence



Recommendation sequence

- Example:
  - Items: videos
  - Steady-state distribution: view counts

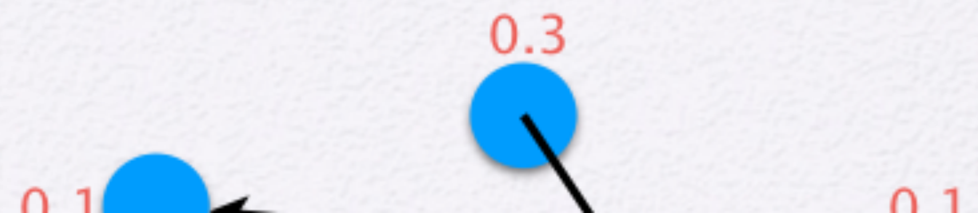
- Why are some videos more popular?

- Why are some videos more popular?
  - Better (higher quality) videos
  - More frequently recommended

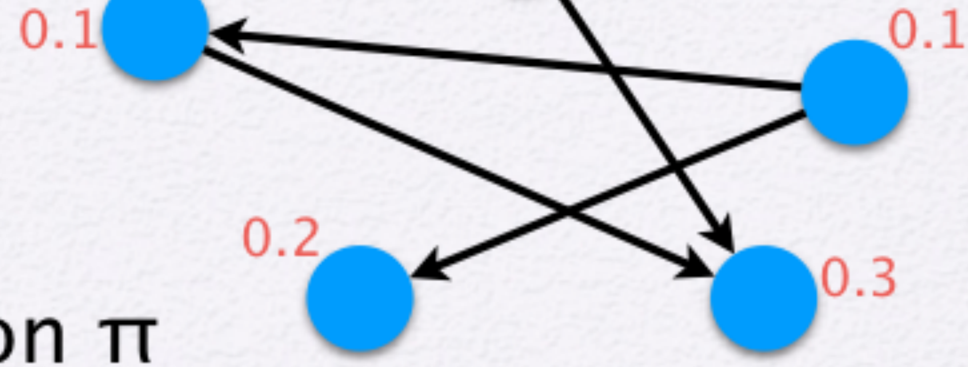
- Would like to disentangle these two reasons
- Given a steady-state, can we infer the quality, in the context of the recommendations?

Inverting a steady state [KumarTVV15]

Given a stationary distribution, find the Markov Chain that generated it



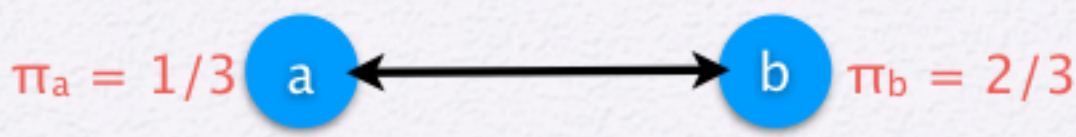
- Directed graph G
  - n nodes
  - m edges
- Steady-state distribution π



Find the transition matrix M that generated π

Feasibility

Not always feasible



A directed graph is consistent if there is a flow that preserves the steady state that preserves the steady state

- Any strongly connected graph with self loops is consistent

Claim. For any consistent graph G, there exists a Markov chain M with π as its stationary distribution

Under constrained?

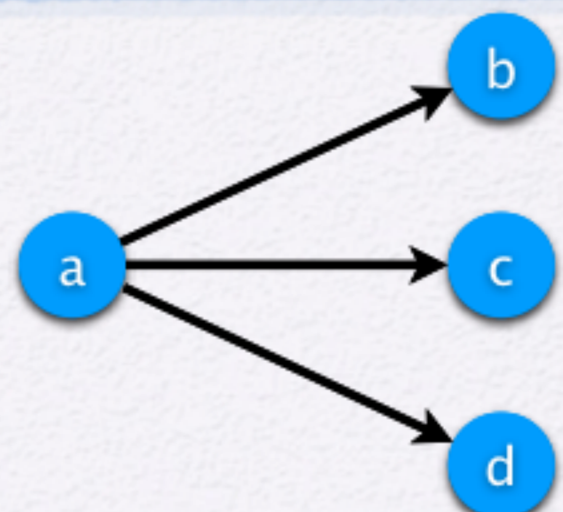
- The problem is under-constrained:
  - n constraints
  - m > n variables
- [Tomlin03]: MaxEnt objective on variables

(regularization)

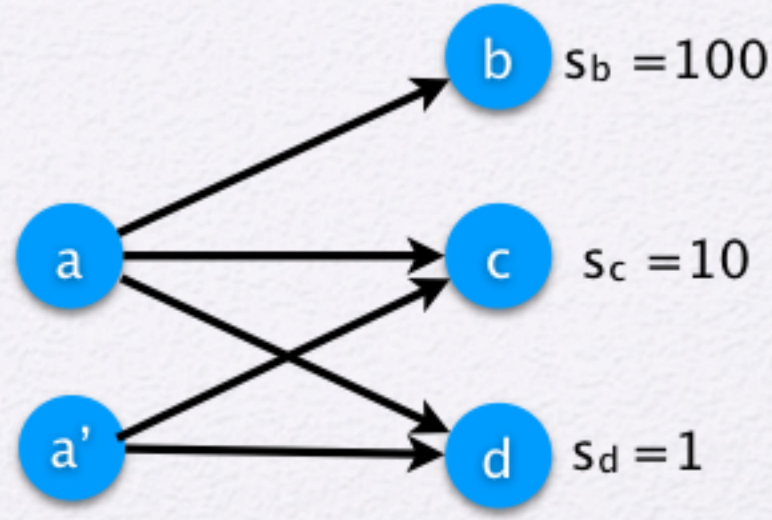
- We will limit the degrees of freedom
  - For each node v_i let s_i be its score
  - Transition matrix M is function of scores
  - Scores express global quality or attractiveness

A simple example

- Transition probability proportional to the score of the node
  - Eg, $M_{ac} = s_c / (s_b + s_c + s_d)$

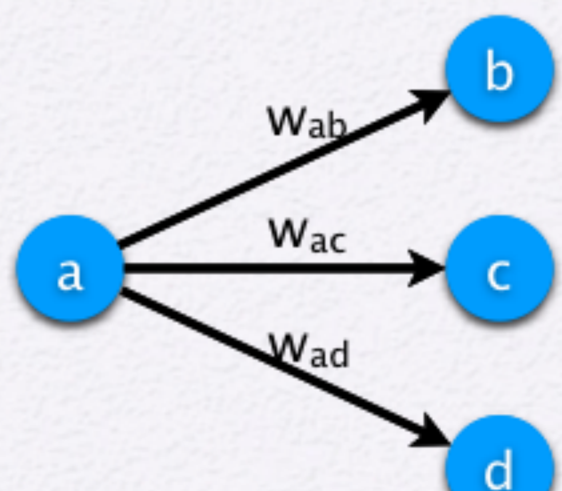


- Transition probabilities are context dependent
  - IIA: Luce Axiom
  - Eg, $M_{ac} = 0.01$, $M_{a'c} = 0.91$



A general formulation

- f(s, w) is a function
  - s is node score
  - w is node-node similarity
- Transition probability



$$M_{ac} = \frac{f(s_c, w_{ac})}{f(s_b, w_{ab}) + f(s_c, w_{ac}) + f(s_d, w_{ad})}$$

- f should be continuous, monotone in s, and unbounded in s, ie, $\lim_{s \to \infty} f(s, w) = \infty$

Reconstruction

Claim. Given a consistent input G and π and a monotone continuous unbounded function f, there is

- a unique set of scores $s_1, \ldots, s_n$
- the transition matrix defined using the scores

and f has π as its stationary distribution

- the scores can be found in polynomial time

- Uniqueness: up to scaling (with assumptions)
- Recovery: scores can be arbitrarily approximated

Main idea

- Fix a set s of scores and target distribution π
- Let $q_i(s)$ be the expected mass at v_i starting with π using s

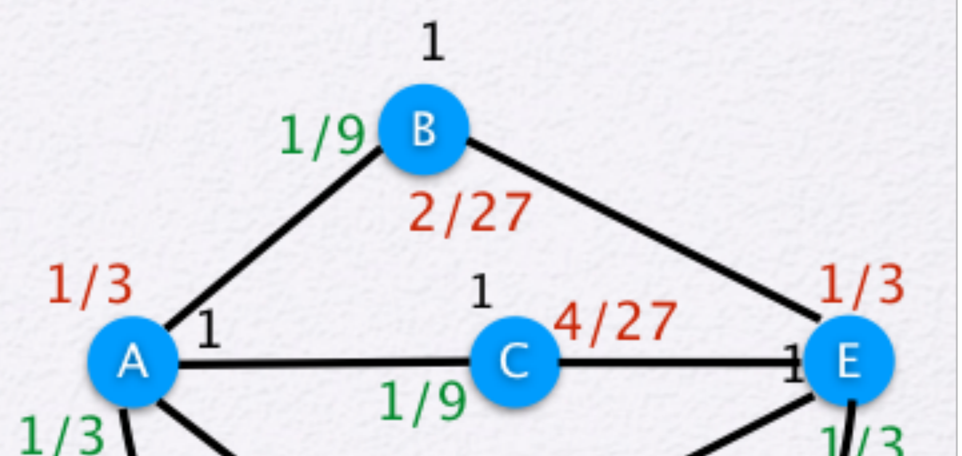
- Call a node v_i underweight if

$$q_i(s) < (1 - \varepsilon)\, \pi_i \qquad 1$$

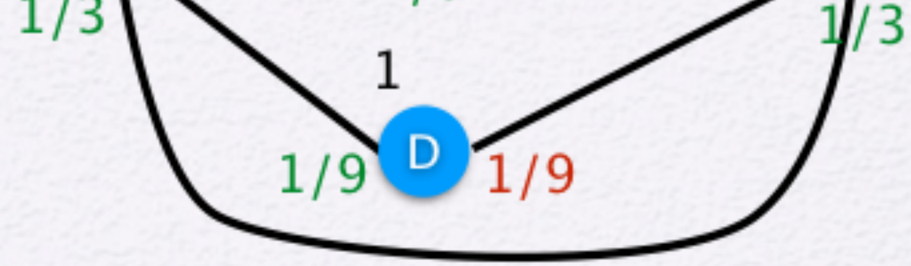
- Repeatedly increase scores of underweight nodes

Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π

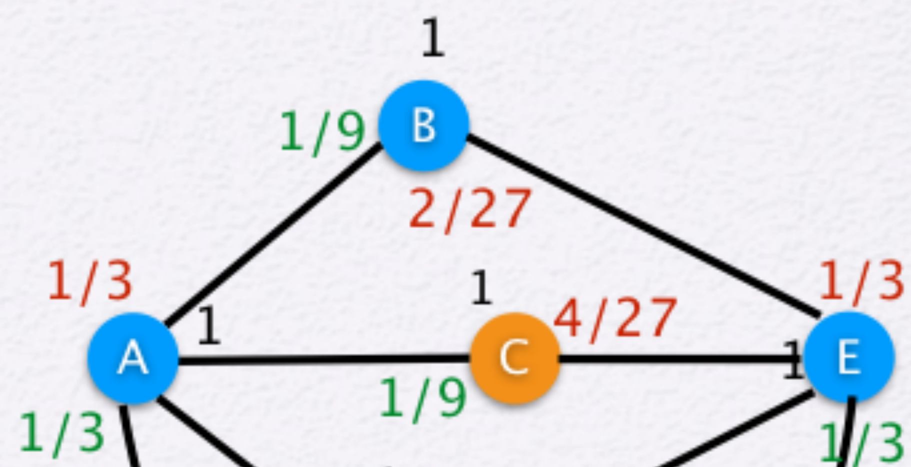


mass at v_i starting with π using s
- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\ \pi_i$



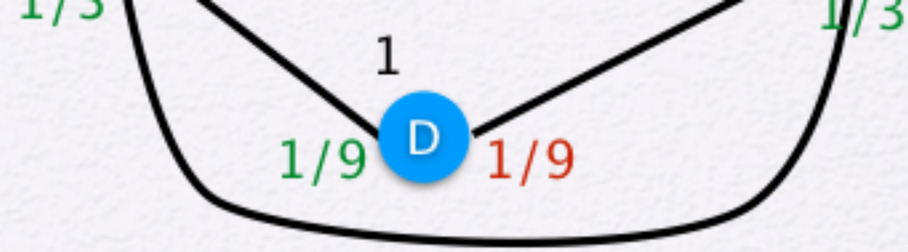
Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π



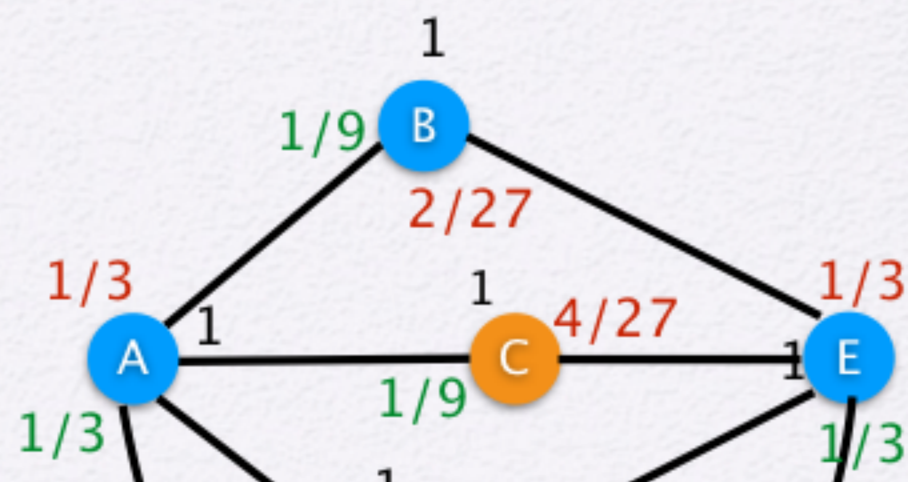
mass at v_i starting with π using s

- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\ \pi_i$



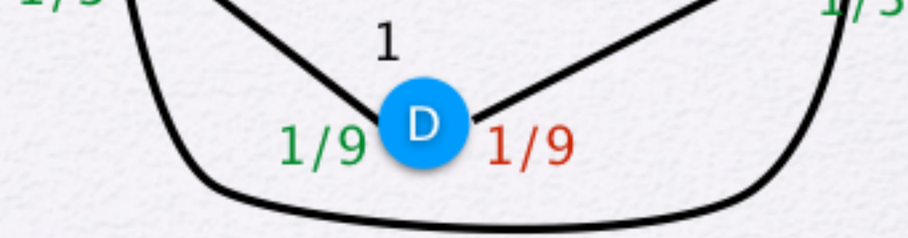
Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π



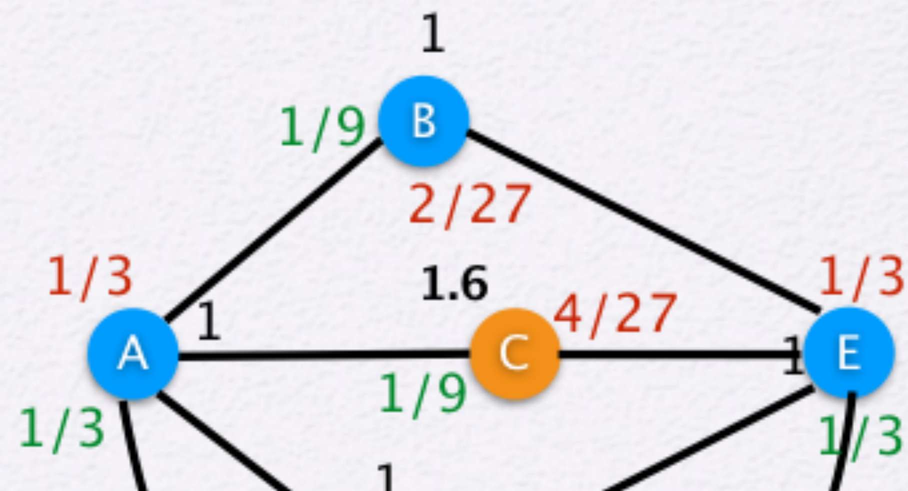
using s

- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\, \pi_i$
- Increase score of underweight nodes

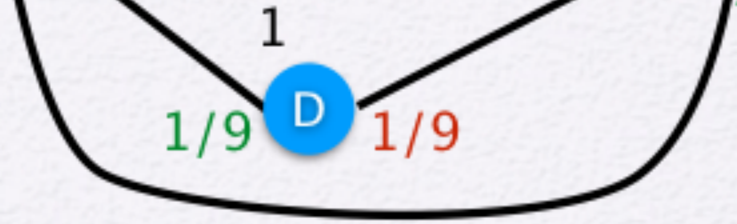


Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π

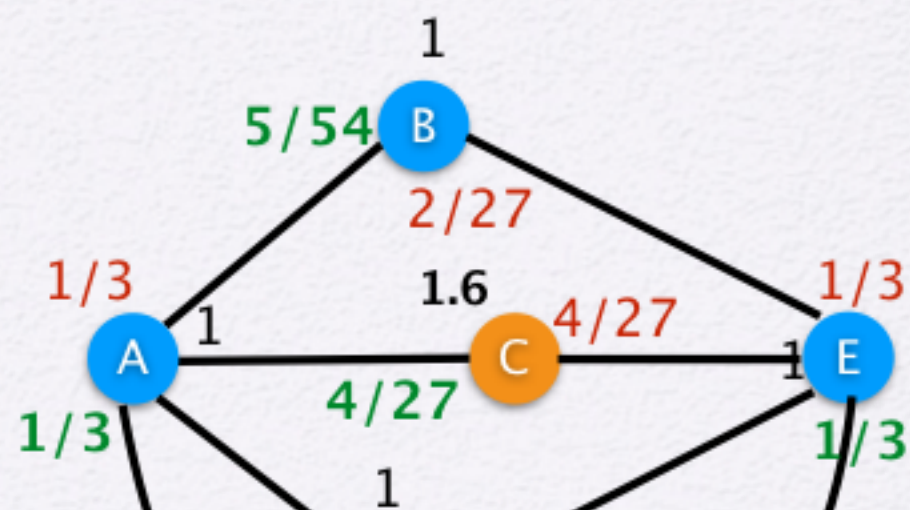


using s
- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\, \pi_i$
- Increase score of underweight nodes



Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π using s

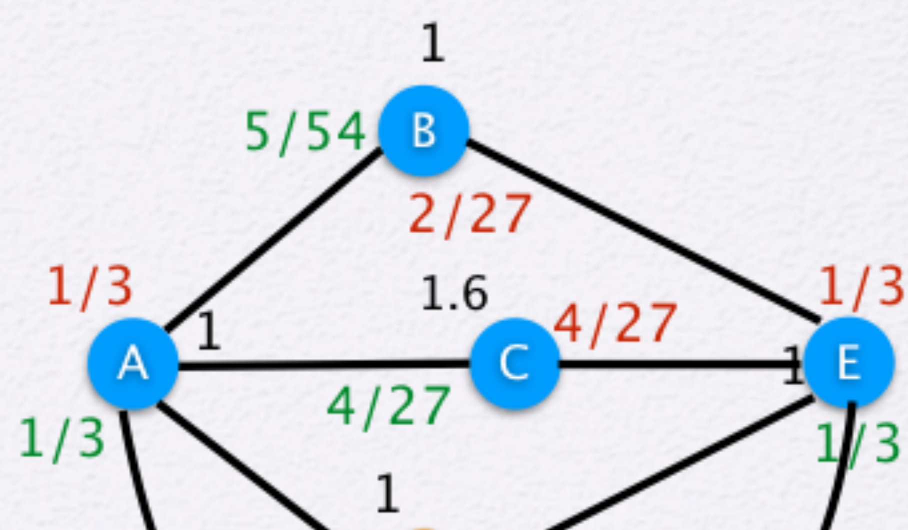


using s

- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\ \pi_i$
- Increase score of underweight nodes

Main idea: Example

- Given a target distribution π
- Fix a set s of scores
- Let $q_i(s)$ be the expected mass at v_i starting with π using s



using s

- Call node v_i underweight if $q_i(s) < (1 - \varepsilon)\, \pi_i$
- Increase score of underweight nodes

Algorithm

Initialize $s_i^0 = 1/n$

for $t = 1, 2, \ldots$
  for each v_i in V
    if v_i is underweight
      find s_i^t such that

Scores never decrease

Guaranteed to exist since f is monotone,

$q_i(s_{-i}^{t-1}, s_i^t) = (1-\varepsilon)\,\pi_i$

else

$s_i^t = s_i^{t-1}$

continuous,
unbounded and
and G is consistent

If q is ever below π it will always stay below

Why does this work?

Claim. There is an explicit bound B such that for all i, t we have $s_i^t < B$

- Consider a set of scores that grows without bound
- These scores all must be underweight (these are the only scores that increase)
- Not all scores can be underweight (sum of

- Not all scores can be underweight (sum of underweight scores below 1)
- The scores growing without bound are taking all of the probability mass from those bounded
- By consistency, this demand must be met, a contradiction

Empirical evaluation

- Dataset: empirical transitions
- Input: Transition graph and the steady state distribution
- Output: Transition probabilities
- Metrics: LogLikelihood or RMSE

Datasets

- Wikitrails:
  - Navigation paths through Wikipedia
  - About 200k transition pairs, 51k user traces over 4.6k nodes
- Restaurants:
  - Results of broad restaurant queries to Google
  - 100k transitions, 65k nodes
- Entree:

- Chicago restaurant recommendation system
- 50k transitions, 27k nodes
- Comedy:
  - Given a video pair, predict which is judged funnier
  - 225k transitions, 75k nodes

Baselines

- Popularity:
  - Transition proportionally to the steady state distribution (score = π)
- Uniform:
  - Uniform over out-edges
- Pagerank:
  - Transition proportionally to the node PageRank
- Temperature:

- Temperature:
  - MaxEnt regularization approach
- Inversion:
  - Our algorithm

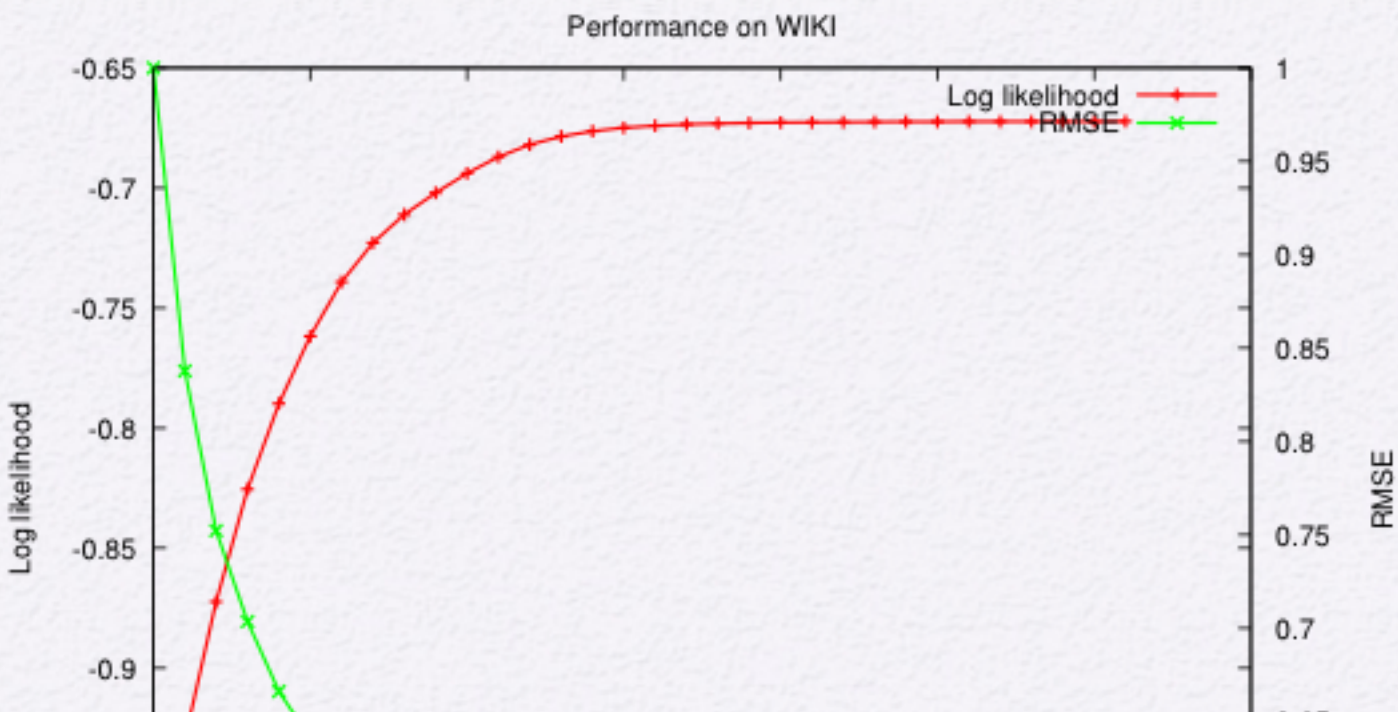
RMSE prediction

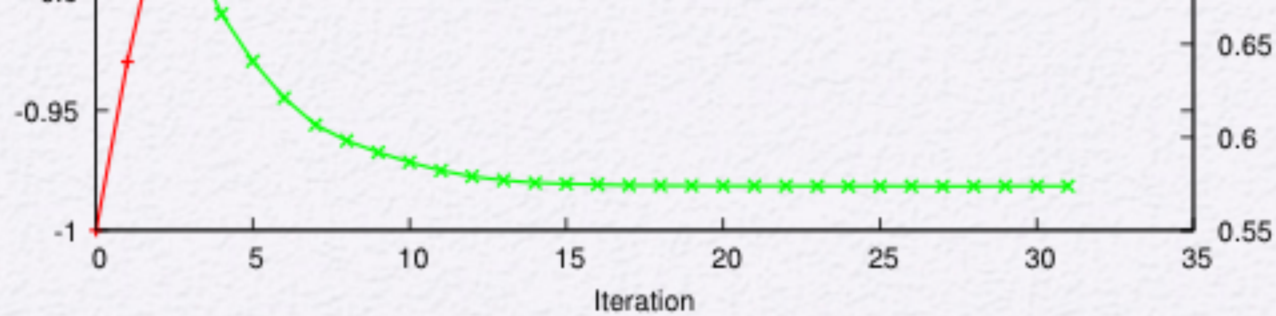
- Expressed as function of Popularity

	Uniform	PageRank	Temp.	Inversion
Wikitrails	0.65	0.83	0.65	0.57
Restaurant	1.17	1.39	1.21	0.59
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Entree	0.69	1.01	0.56	0.42
Comedy	0.65	0.9	0.78	0.36

Convergence

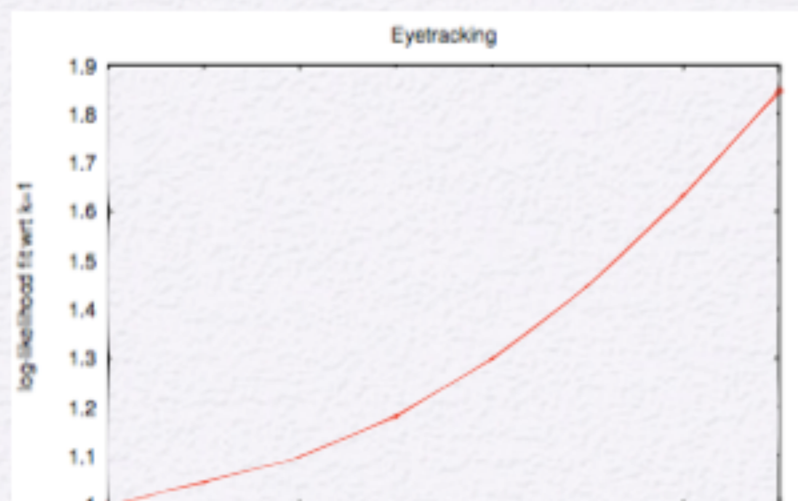


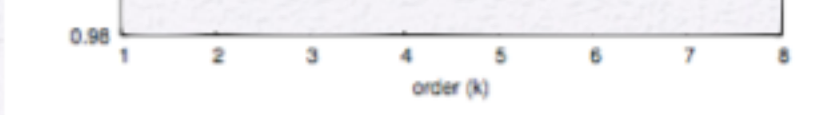


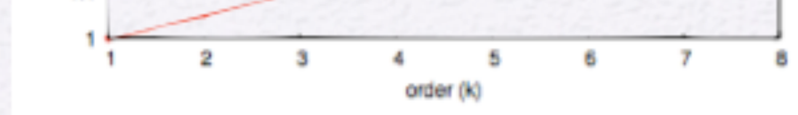
Are sequences Markovian?

[ChierichettiKRT12]

- k-th order Markov process: next state depends on the previous k states







- Expensive to compute and store
- Data sparsity
- Can compress into a variable-order chain

Linear-Additive Markov Process [KumarRST16]

- LAMP derives inspiration from recency aspect in repeat consumption

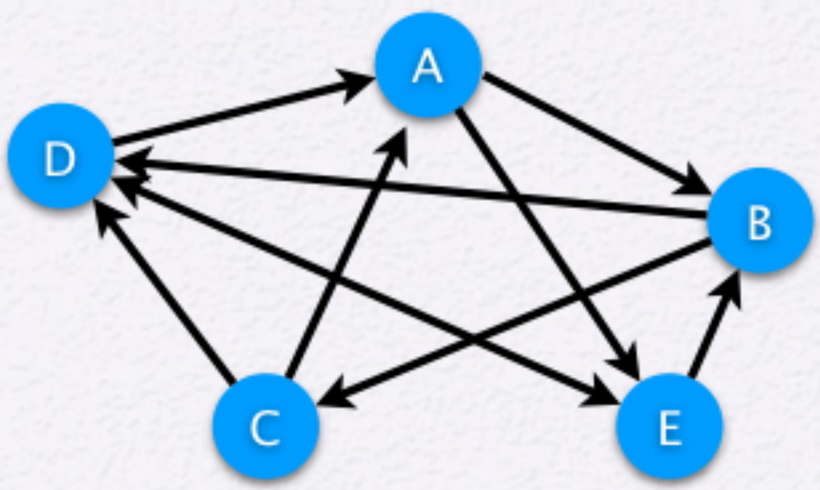
Given a Markov chain M and a weight vector w of length k

Suppose current sequence is $x_1, \ldots, x_l$

- Flip a coin according to w to get an index

- Flip a coin according to w to get an index $i \in [l-k, l]$
- Go back to x_i in the sequence and use M to transition from x_i

LAMP: Example



Transition matrix
weight = (1/2, 1/2)

A E E B B C D A A …

Captures some aspect of long-range dependencies

Power of LAMP

- More powerful than standard Markov chains
- Less powerful than higher-order Markov chains
- More succinct representation
- Has nice theoretical properties
  - Connections to renewal processes
- Has an efficient estimation algorithm
  - Non-convex problem

- Alternating maximization

Concluding thoughts

- Behavioral sequences have a lot of usable information
- Simple models can be used to explain the data
  - Standard tools from statistics/learning
  - Value in analysis
- Insights from these can be used to improve

- Insights from these can be used to improve end user experience

Thank you!

Questions/Comments

ravi.k53 @ gmail