Web Mining and Recommender Systems

Temporal data mining: Regression for Sequence Data

Learning Goals

- Discuss how to use regression to predict temporally evolving data

This topic

Temporal models

This topic will look back on some of the topics already covered in this class, and see how they can be adapted to make use of **temporal** information

1. **Regression** – sliding windows and autoregression
2. **Social networks** – densification over time
3. **Text mining** – "Topics over Time"
4. **Recommender systems** – some results from Koren

Previously – Regression

Given **labeled training data** of the form

$$\{(\text{data}_1, \text{label}_1), \ldots, (\text{data}_n, \text{label}_n)\}$$

Infer the function

$$f(\text{data}) \stackrel{?}{\rightarrow} \text{labels}$$

Time-series regression

Here, we'd like to predict sequences of **real-valued** events as accurately as possible.

$x_1 \; . \; . \; . \; . \; x_n \to x_{n+1} \to x_{n+2}$

data

label

label

data

Time-series regression

Here, we'd like to predict sequences of **real-valued** events as accurately as possible.

Given: a time series:

$$(x_1, \ldots, x_N) \in \mathbb{R}^N$$

Suppose we'd like to minimize the MSE (as usual!) of the final part of some continuous portion of the sequence

$$\tfrac{1}{u-v+1} \sum_{t=u}^{v} (f_t(x_1, \ldots, x_{u-1}) - x_t)^2$$

Method 1: maintain a "moving average" using a window of some fixed length

$$f(x_1, \ldots, x_m) = \quad (x_m + x_{m-1} + \cdots + x_{m-K+1}) / K$$

$$\frac{\sum_{k=0}^{K-1} x_{m-k}}{K}$$

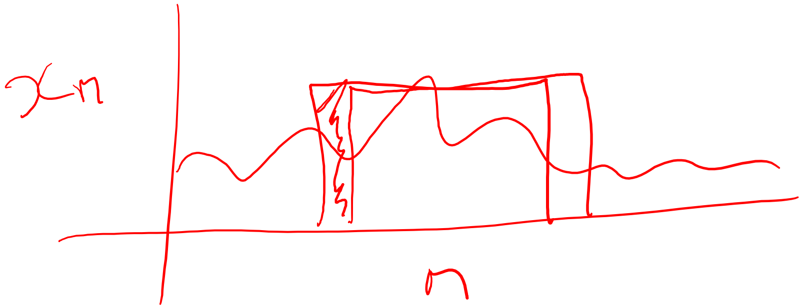
$$O(mK)$$

Time-series regression

Method 1: maintain a "moving average" using a window of some fixed length

- This can be computed efficiently via dynamic programming:

$$f(x_1, \ldots, x_{m+1}) = \frac{K f(x_1, \ldots x_m) - x_{m-K+1} + x_{m+1}}{K}$$



Time-series regression

Method 1: maintain a “moving average” using a window of some fixed length

$$f(x_1, \ldots, x_m) = \frac{1}{K} \sum_{k=0}^{K-1} x_{m-k}$$

- This can be computed efficiently via dynamic programming:

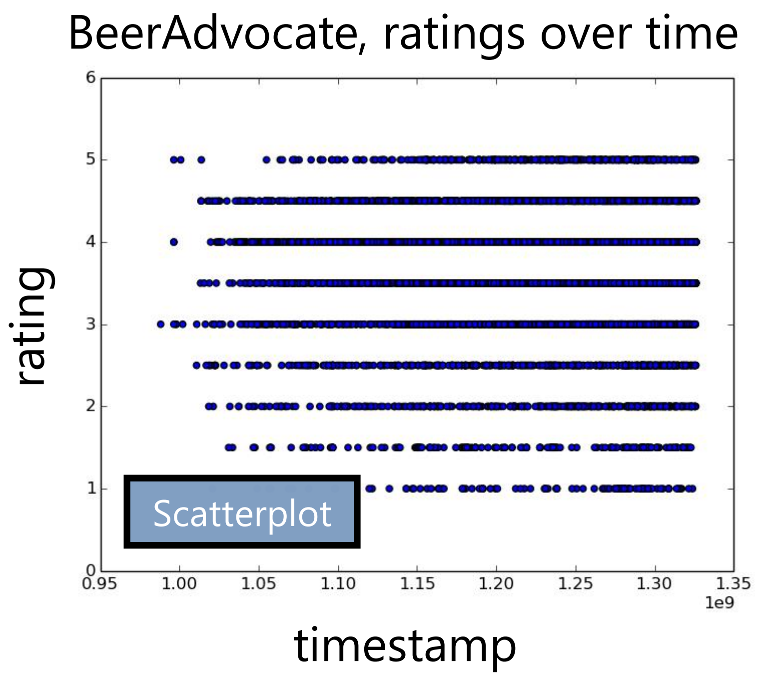
$$f(x_1, \ldots, x_{m+1}) = \frac{1}{K}(K \cdot f(x_1, \ldots, x_m) - x_{m-k} + x_{m+1})$$

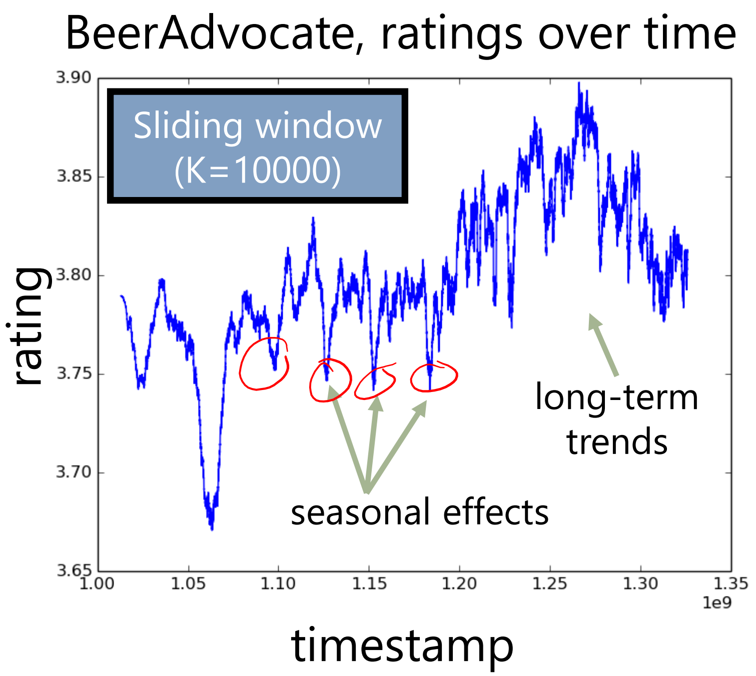
“peel-off” the oldest point

add the newest point

Time-series regression

Also useful to plot data:





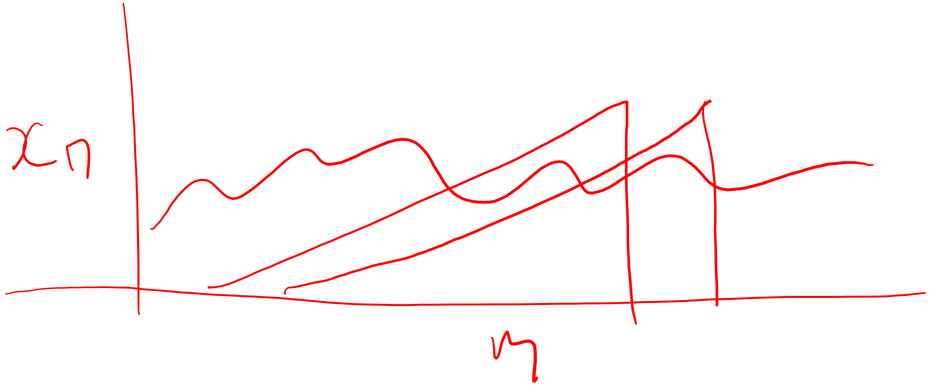
Code on course webpage

Time-series regression

Method 2: weight the points in the moving average by age

$$f(x_1, \ldots, x_m) = \frac{K x_n + (K-1)x_{n-1} + \ldots + 1 \cdot x_{n-K+1}}{1 + 2 + \ldots + K}$$

$$= \frac{\sum_{k=0}^{K-1} (K-k) x_{n-k}}{\binom{K}{2}}$$



Method 2: weight the points in the moving average by age

newest points have the highest weight

weight decays to zero after K points

$$f(x_1, \ldots, x_m) = \frac{\sum_{k=0}^{K-1} (K-k) x_{m-k}}{\binom{K}{2}}$$

Method 3: weight the most recent points exponentially higher

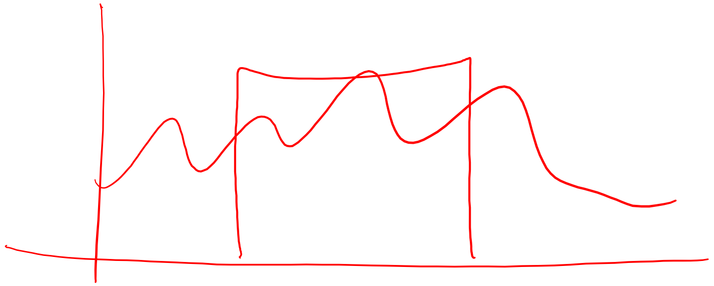
$$f(x_1) = \quad x_1$$

$$f(x_1, \ldots, x_m) = \quad \alpha f(x_1 \ldots x_{m-1}) + (1-\alpha) x_m$$

Methods 1, 2, 3

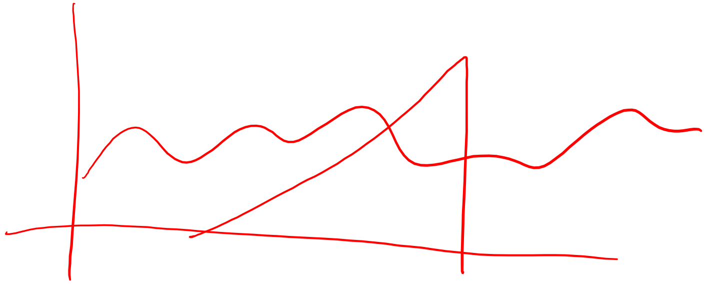
Method 1: Sliding window
Method 2: Linear decay
Method 3: Exponential decay

1

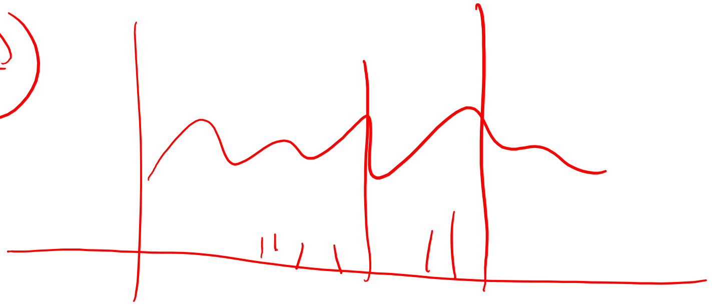


3

2



4



Method 4: all of these models are assigning **weights** to previous values using some predefined scheme, why not just **learn** the weights?

$$f(x_1, \ldots, x_m) = \theta_0 x_m + \theta_1 x_{n-1} + \ldots \theta_{k-1} x_{n-k+1}$$

$$\sum_{k=0}^{k-1} \theta_k x_n$$

$$\sum_{m} \left(f(x_1 + \ldots x_n) - x_{n-1} \right)^2$$

Time-series regression

Method 4: all of these models are assigning **weights** to previous values using some predefined scheme, why not just **learn** the weights?

- We can now fit this model using least-squares
- This procedure is known as **autoregression**
- Using this model, we can capture **periodic** effects, e.g. that the traffic of a website is most similar to its traffic 7 days ago

Learning Outcomes

- Introduced several schemes to predict values in sequences
- Introduced autoregression

Web Mining and Recommender Systems

Temporal dynamics in social networks

Learning Goals

- Discuss how social networks change over time

Previously…

How can we **characterize, model,** and **reason about** the structure of social networks?

1. Models of network structure
2. Power-laws and scale-free networks, “rich-get-richer” phenomena
3. Triadic closure and “the strength of weak ties”
4. Small-world phenomena
5. Hubs & Authorities; PageRank

Temporal dynamics of social networks

Previously we saw some processes that model the generation of social and information networks

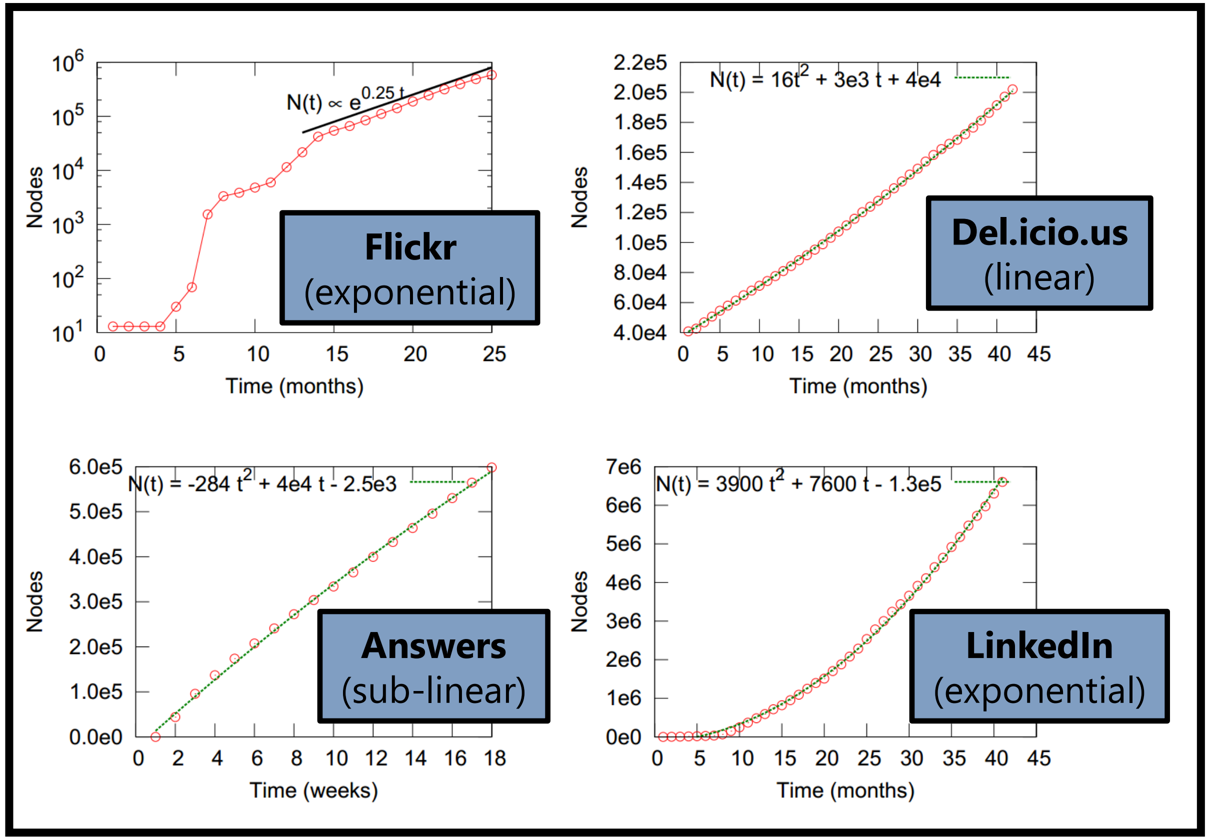
- Power-laws & small worlds
- Random graph models

These were all defined with a “static” network in mind. But if we observe the **order** in which edges were created, we can study how these phenomena change as a function of time

First, let’s look at “microscopic” evolution, i.e., evolution in terms of individual nodes in the network

Temporal dynamics of social networks

Q1: How do networks grow in terms of the number of nodes over time?



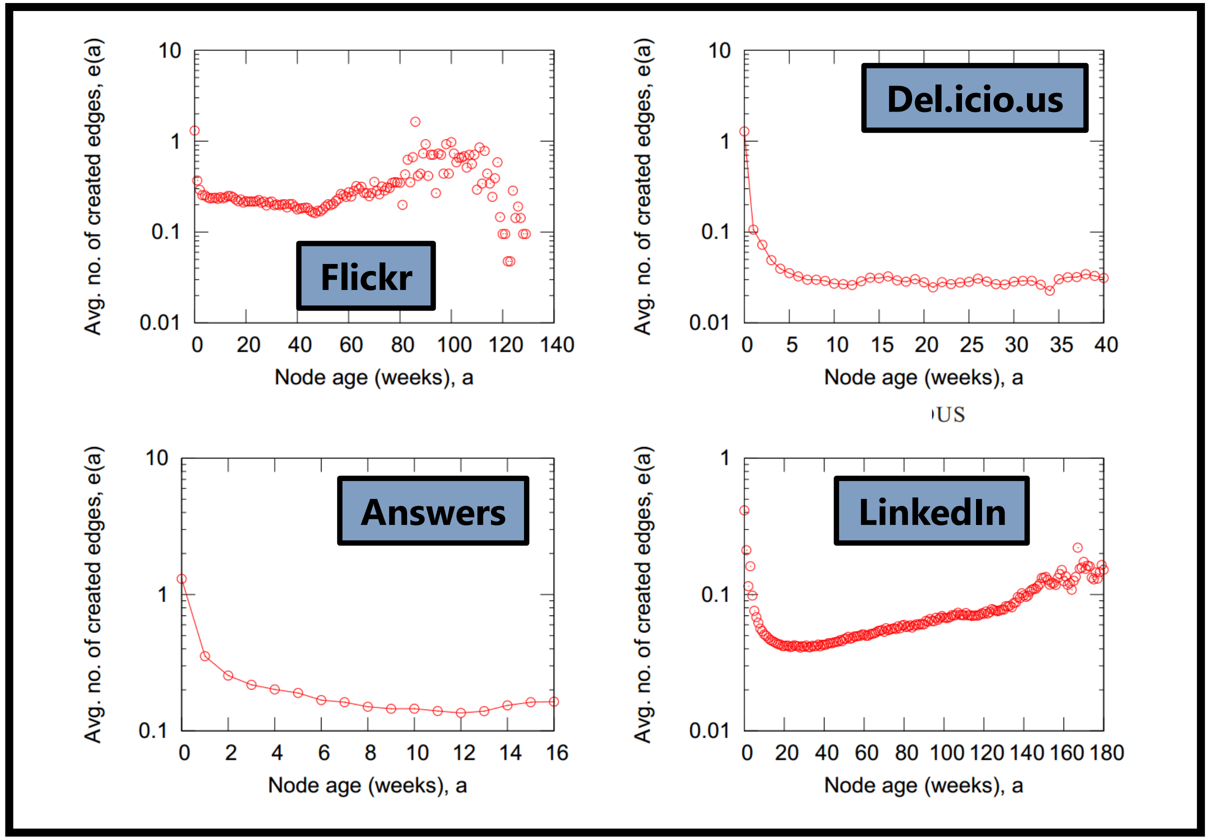
A: Doesn't seem to be an obvious trend, so what **do** networks have in common as they evolve?

(from Leskovec, 2008 (CMU Thesis))

Temporal dynamics of social networks

Q2: When do nodes create links?

- x-axis is the age of the nodes
- y-axis is the number of edges created at that age

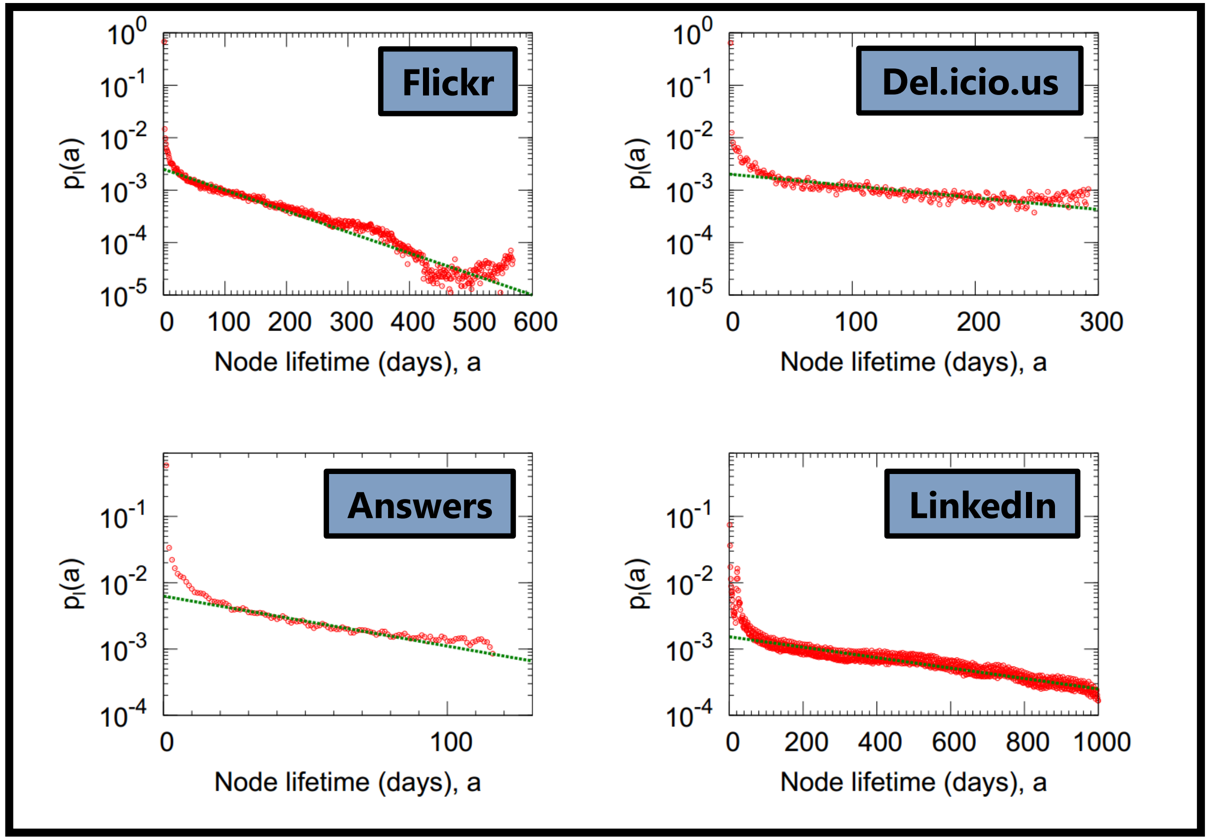


A: In most networks there's a "burst" of initial edge creation which gradually flattens out. Different behavior on LinkedIn?

Temporal dynamics of social networks

Q3: How long do nodes "live"?

- x-axis is the diff. between date of last and first edge creation
- y-axis is the frequency

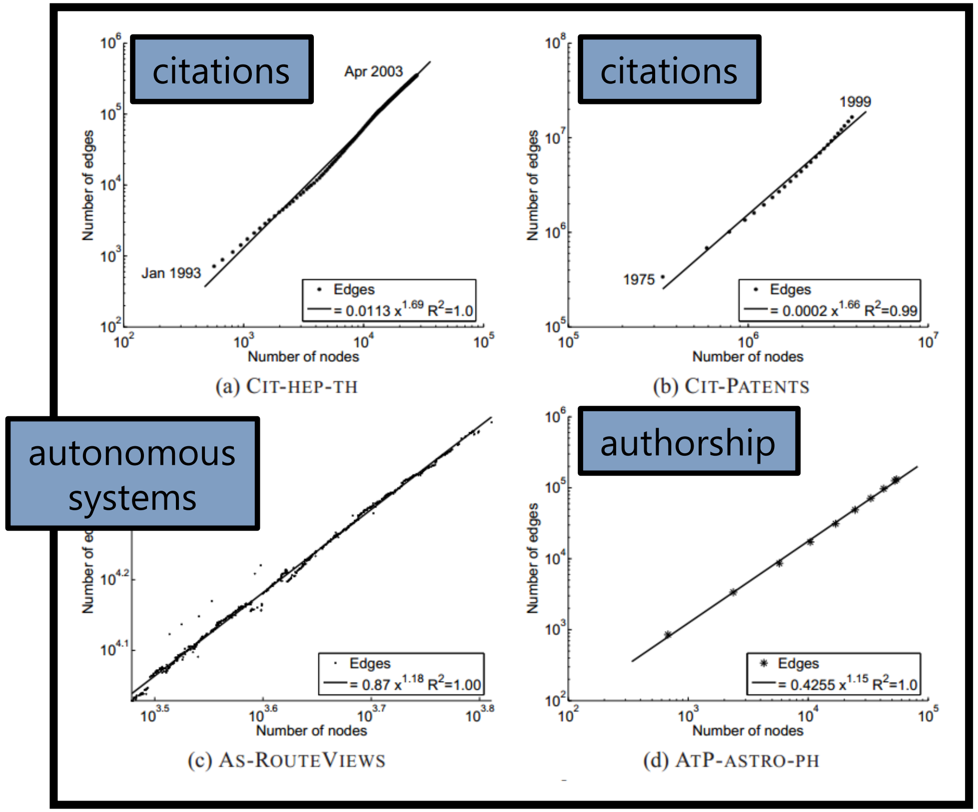


A: Node lifetimes follow a power-law: many many nodes are shortlived, with a long-tail of older nodes

Temporal dynamics of social networks

What about "macroscopic" evolution, i.e., how do global properties of networks change over time?

Q1: How does the # of nodes relate to the # of edges?



- A few more networks: citations, authorship, and autonomous systems (and some others, not shown)
- **A:** Seems to be linear (on a log-log plot) **but** the number of edges grows **faster** than the number of nodes as a function of time

Temporal dynamics of social networks

Q1: How does the # of nodes relate to the # of edges?
A: seems to behave like

$$E(t) \propto N(t)^a$$

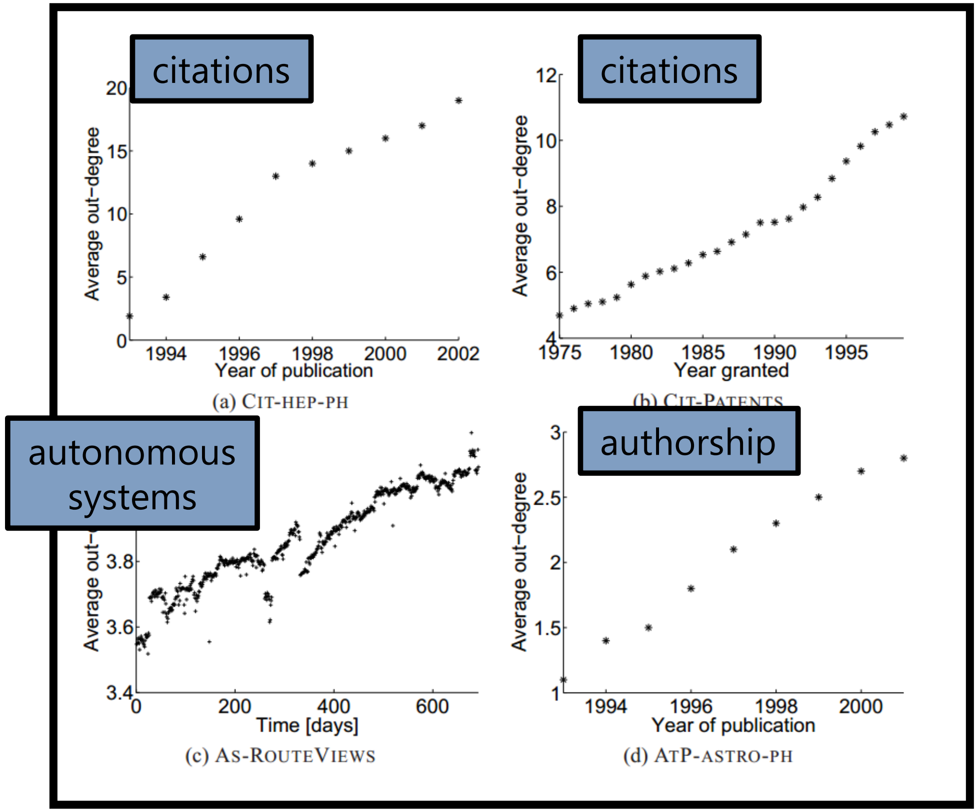
where

$$1 \leq a \leq 2$$

- a = 1 would correspond to **constant** out-degree – which is what we might traditionally assume
- a = 2 would correspond to the graph being fully connected
- What seems to be the case from the previous examples is that a > 1 – the number of edges grows faster than the number of nodes

Temporal dynamics of social networks

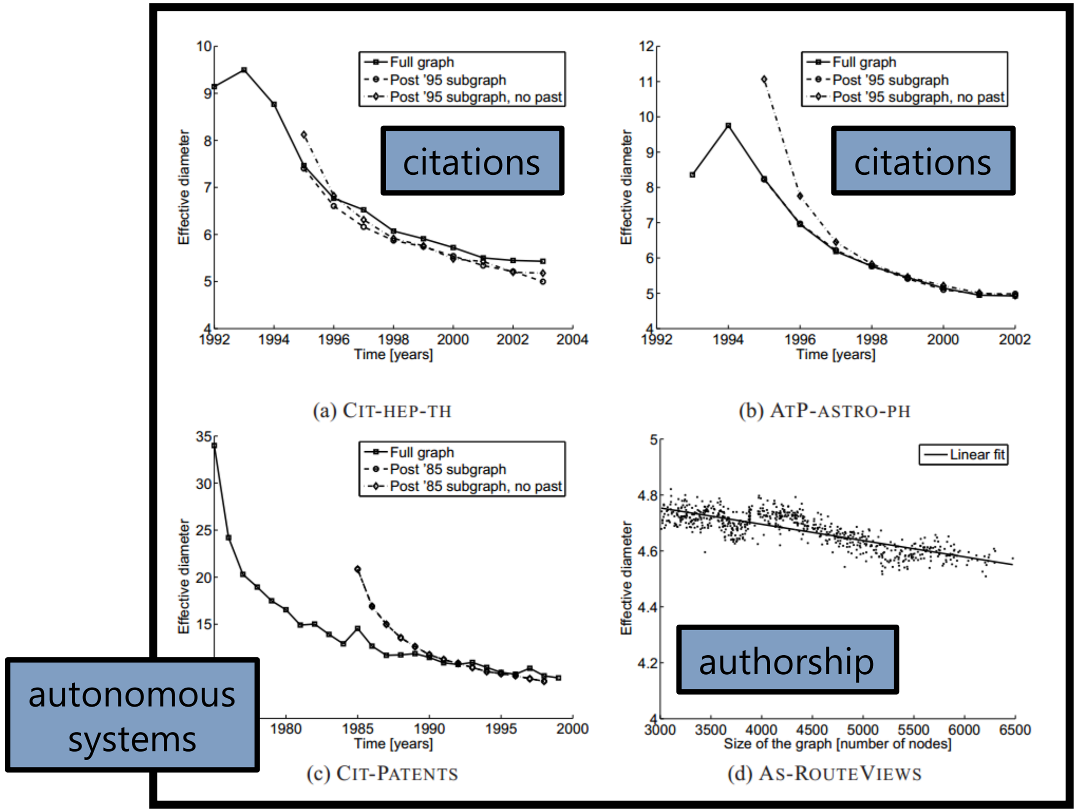
Q2: How does the degree change over time?



- **A:** The average out-degree **increases** over time

Temporal dynamics of social networks

Q3: If the network becomes **denser**, what happens to the (effective) diameter?

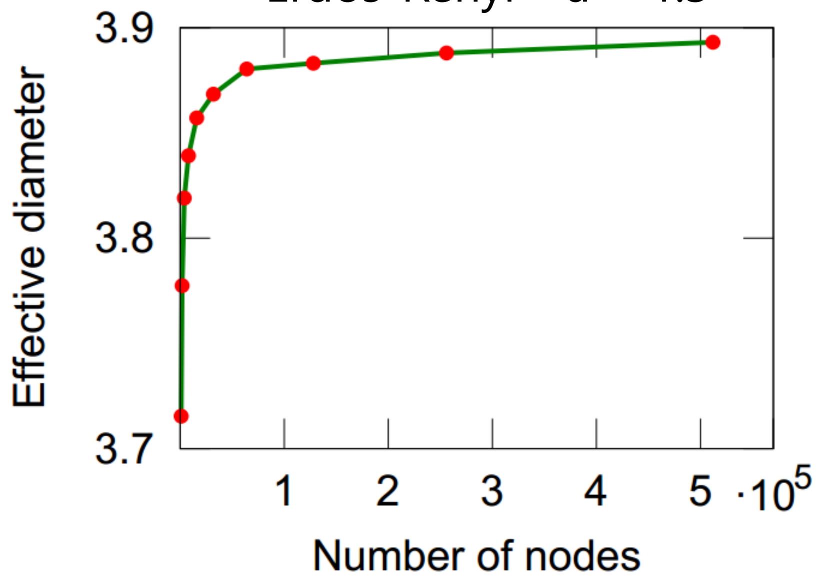


- **A:** The diameter seems to decrease
- In other words, the network becomes **more** of a small world as the number of nodes increases

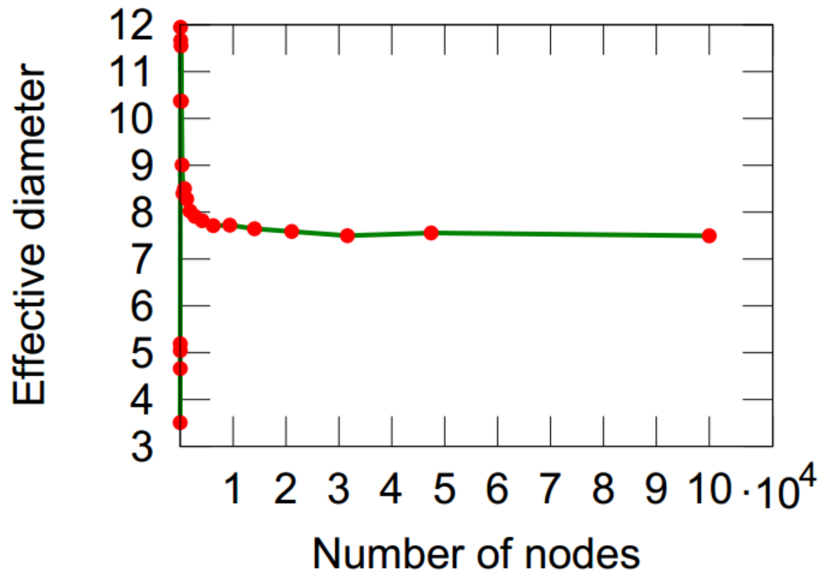
Temporal dynamics of social networks

Q4: Is this something that **must** happen – i.e., if the number of edges increases faster than the number of nodes, does that mean that the diameter must decrease?
A: Let's construct random graphs (with a > 1) to test this:

Erdos-Renyi – a = 1.3



Pref. attachment model – a = 1.2

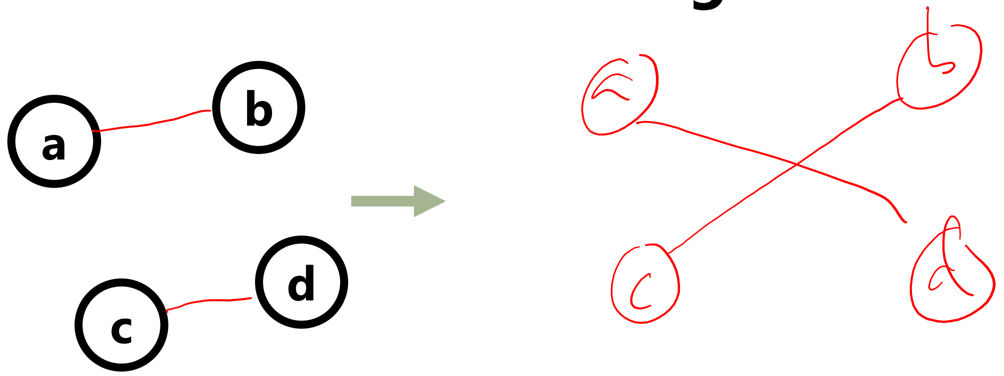


Temporal dynamics of social networks

So, a decreasing diameter is **not** a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model

Q5: is the degree distribution of the nodes sufficient to explain the observed phenomenon?

A: Let’s perform **random rewiring** to test this

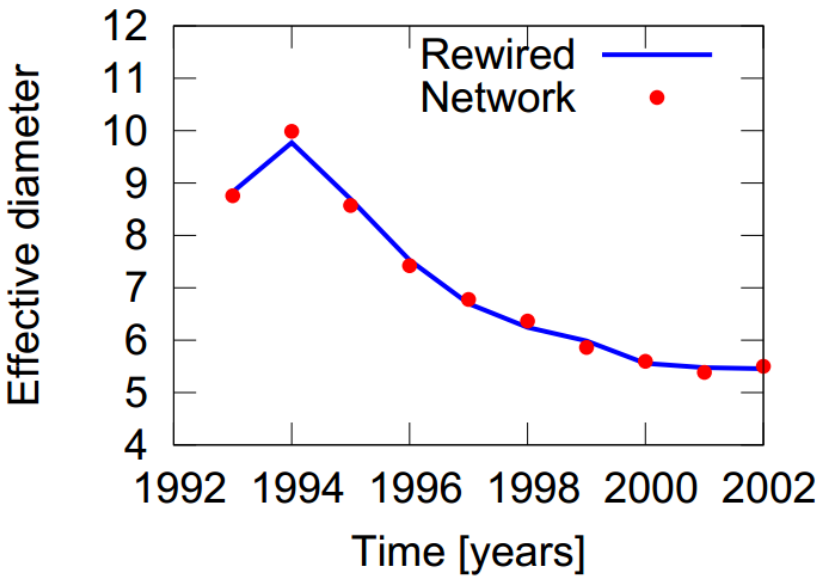


random rewiring preserves the degree distribution, and randomly samples amongst networks with observed degree distribution

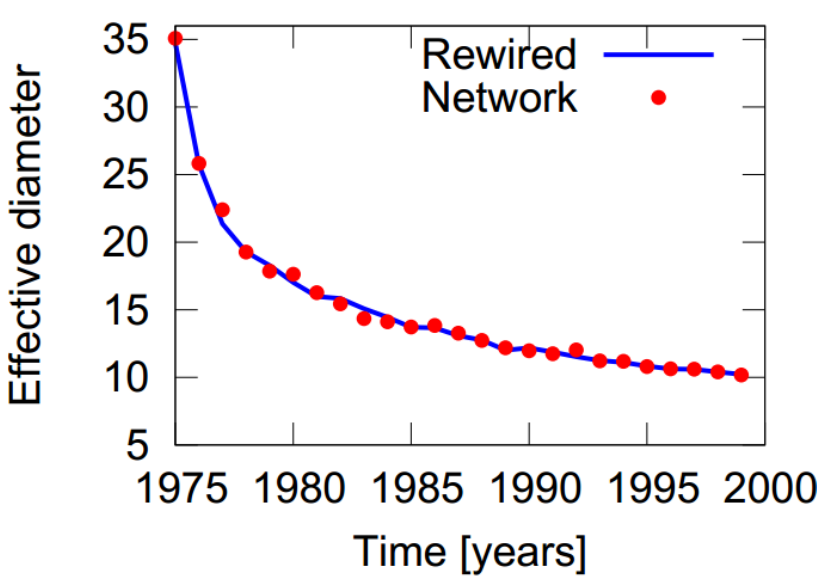
Temporal dynamics of social networks

So, a decreasing diameter is **not** a "rule" of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model

Q5: is the degree distribution of the nodes sufficient to explain the observed phenomenon?



(c) Affiliation network (ATP-ASTRO-PH)



(d) US patent citation network (CIT-PATENTS)

Temporal dynamics of social networks

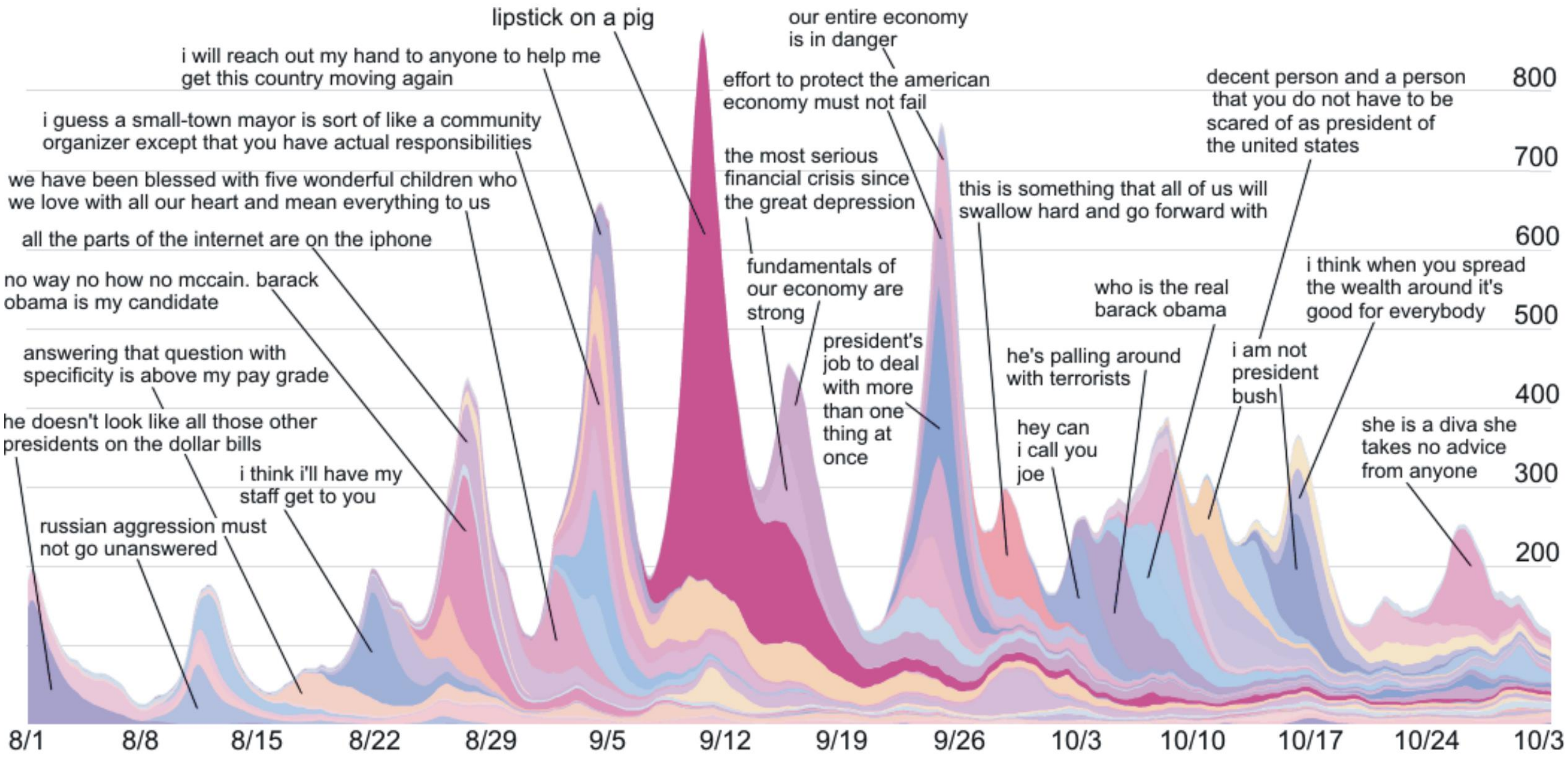
So, a decreasing diameter is **not** a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model

Q5: is the degree distribution of the nodes sufficient to explain the observed phenomenon?

A: Yes! The fact that real-world networks seem to have decreasing diameter over time can be explained as a result of their degree distribution **and** the fact that the number of edges grows faster than the number of nodes

Temporal dynamics of social networks

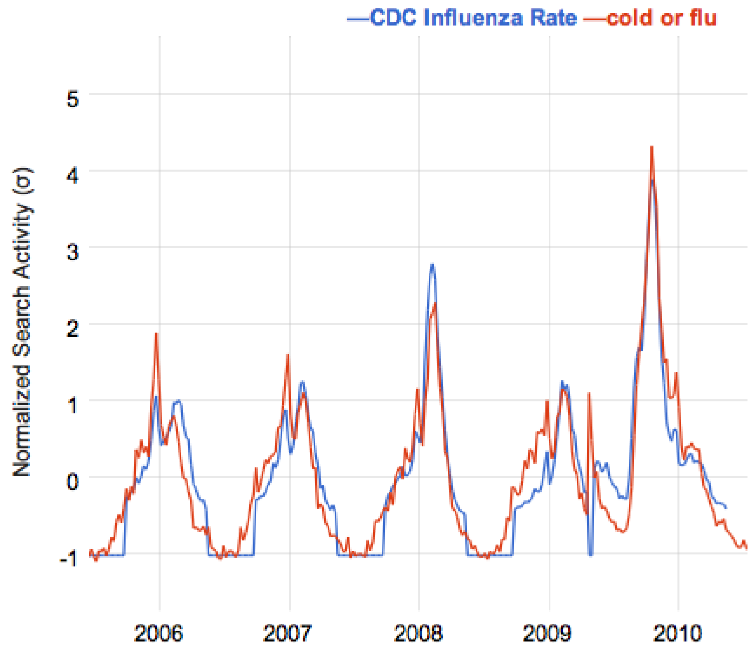
Other interesting topics…



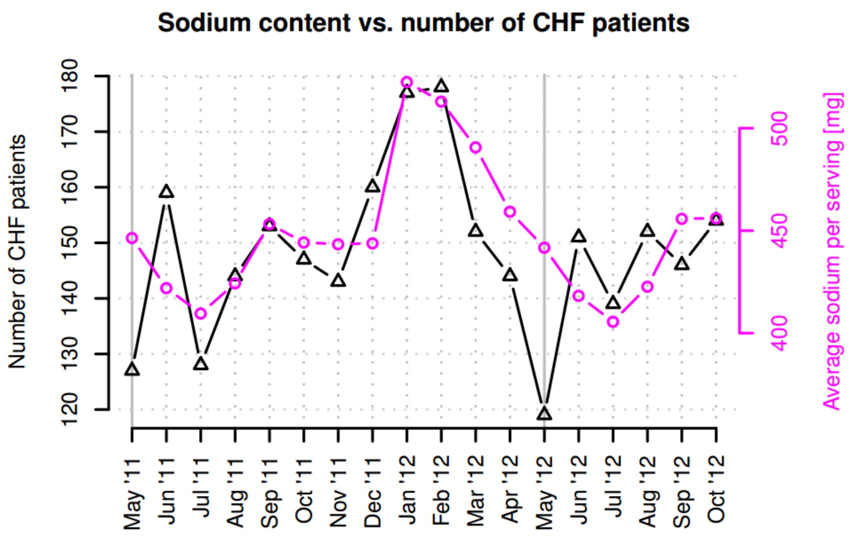
"memetracker"

Temporal dynamics of social networks

Other interesting topics…



Aligning query data with disease data – Google flu trends: https://www.google.org/flutrends/us/#US



Sodium content in recipe searches vs. # of heart failure patients – "From Cookies to Cooks" (West et al. 2013): http://infolab.stanford.edu/~west1/pubs/West-White-Horvitz_WWW-13.pdf

Learning Outcomes

- Discussed how social networks change over time
- Described some mechanisms to explain this phenomenon

References

Further reading:

"Dynamics of Large Networks" (most plots from here)

Jure Leskovec, 2008

http://cs.stanford.edu/people/jure/pubs/thesis/jure-thesis.pdf

"Microscopic Evolution of Social Networks"

Leskovec et al. 2008

http://cs.stanford.edu/people/jure/pubs/microEvol-kdd08.pdf

"Graph Evolution: Densification and Shrinking Diameters"

Leskovec et al. 2007

http://cs.stanford.edu/people/jure/pubs/powergrowth-tkdd.pdf

Web Mining and Recommender Systems

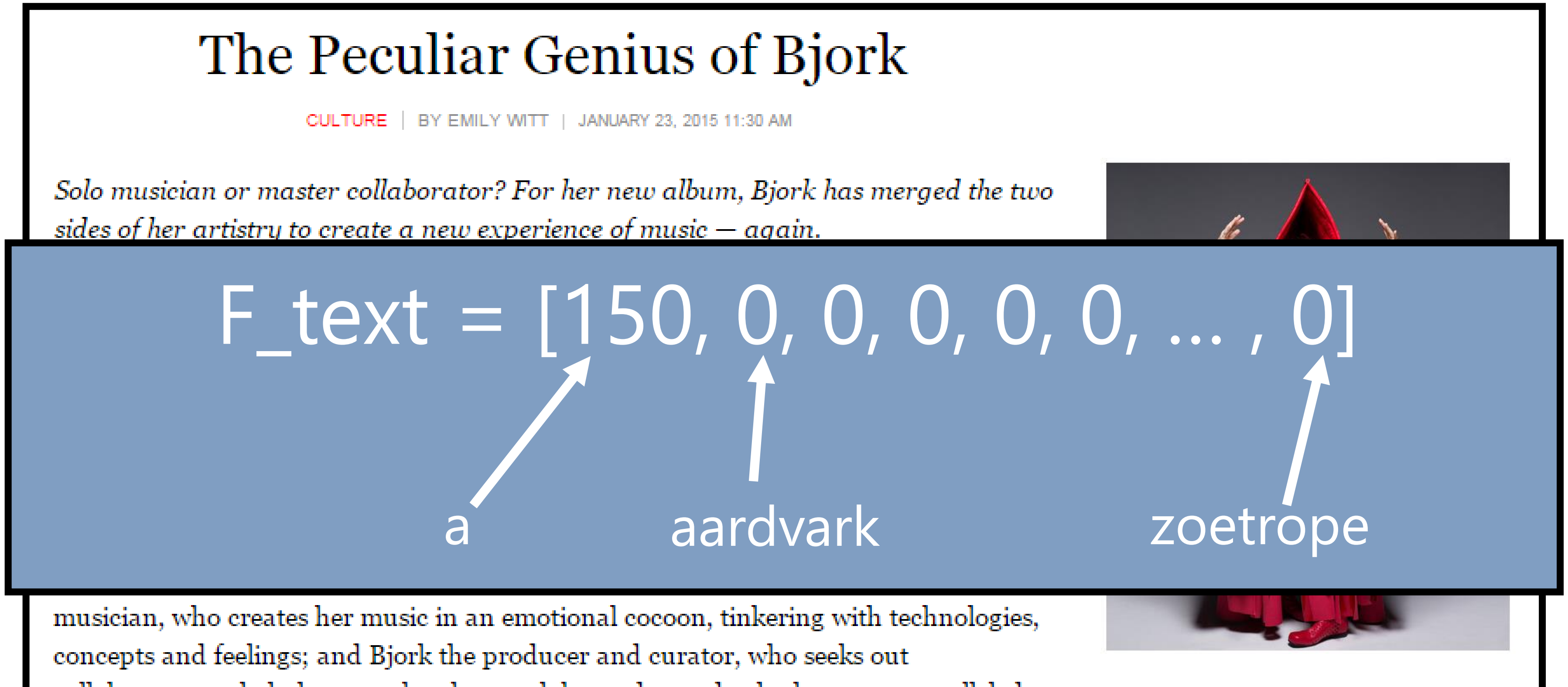
Temporal dynamics of text

Learning Goals

- Discuss how text can change over time

Previously...

Bag-of-Words representations of text:


The Peculiar Genius of Bjork

CULTURE | BY EMILY WITT | JANUARY 23, 2015 11:30 AM

Solo musician or master collaborator? For her new album, Bjork has merged the two sides of her artistry to create a new experience of music — again.

musician, who creates her music in an emotional cocoon, tinkering with technologies, concepts and feelings; and Bjork the producer and curator, who seeks out

F_text = [150, 0, 0, 0, 0, 0, ... , 0]

a

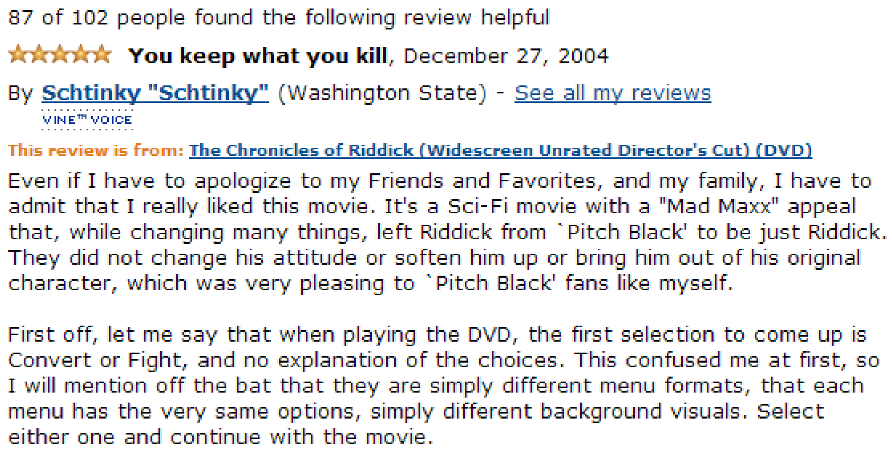
aardvark

zoetrope

Latent Dirichlet Allocation

Previously, we tried to develop low-dimensional representations of documents:

What we would like:


87 of 102 people found the following review helpful

★★★★★ **You keep what you kill**, December 27, 2004

By **Schtinky "Schtinky"** (Washington State) - See all my reviews

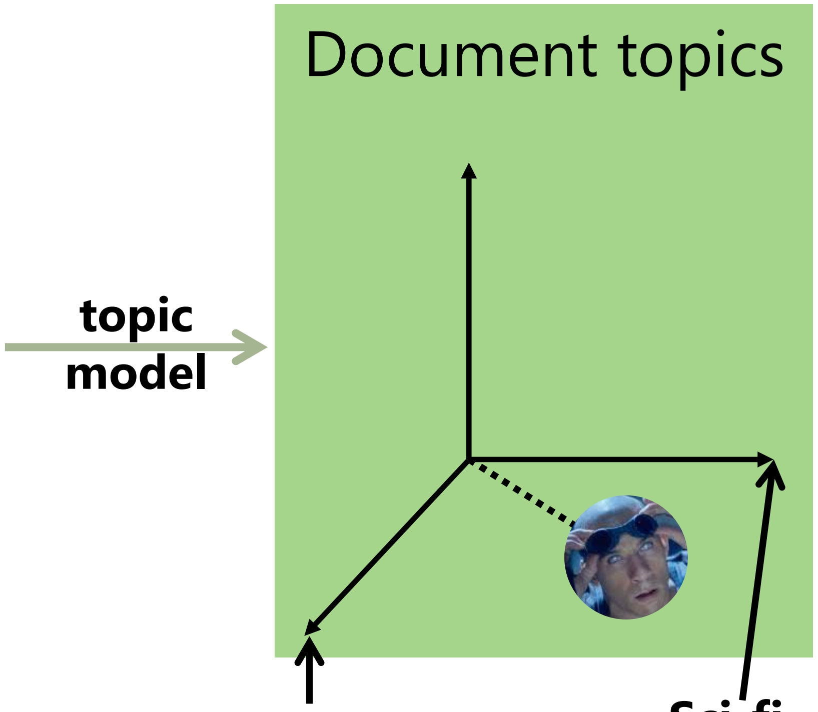
VINE™ VOICE

This review is from: The Chronicles of Riddick (Widescreen Unrated Director's Cut) (DVD)

Even if I have to apologize to my Friends and Favorites, and my family, I have to admit that I really liked this movie. It's a Sci-Fi movie with a "Mad Maxx" appeal that, while changing many things, left Riddick from `Pitch Black' to be just Riddick. They did not change his attitude or soften him up or bring him out of his original character, which was very pleasing to `Pitch Black' fans like myself.

First off, let me say that when playing the DVD, the first selection to come up is Convert or Fight, and no explanation of the choices. This confused me at first, so I will mention off the bat that they are simply different menu formats, that each menu has the very same options, simply different background visuals. Select either one and continue with the movie.

(review of "The Chronicles of Riddick")

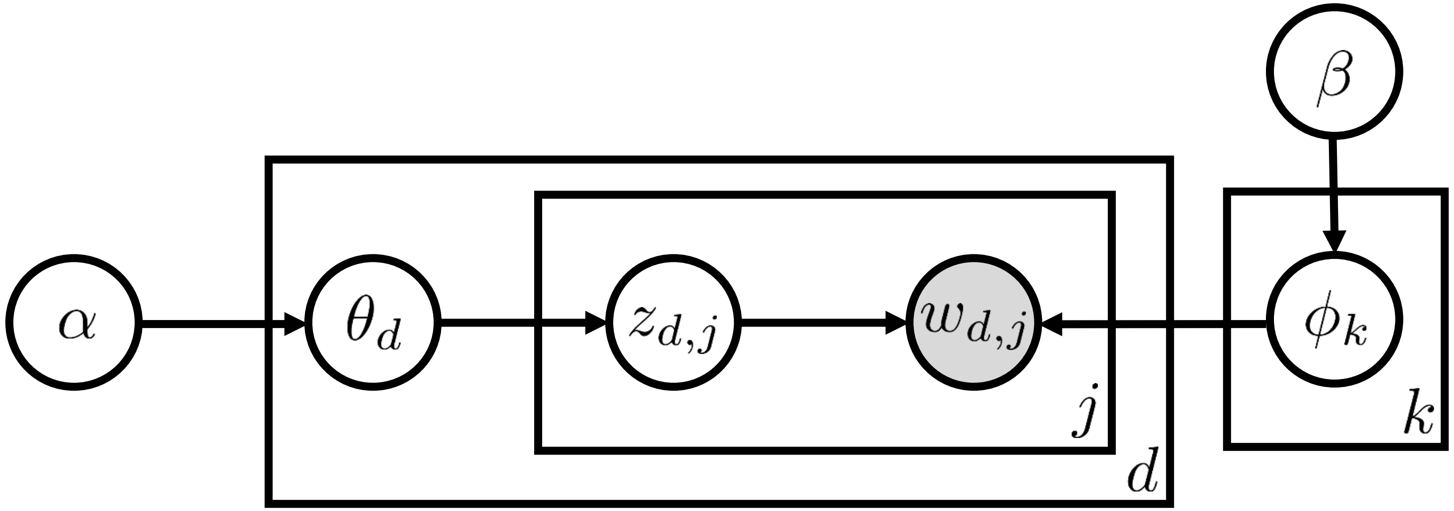


Action:
action, loud, fast, explosion,...

Sci-fi
space, future, planet,...

Latent Dirichlet Allocation

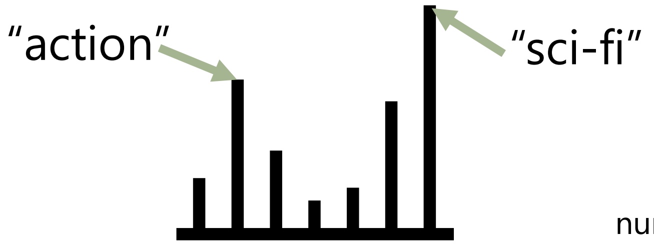
We saw how **LDA** can be used to describe documents in terms of **topics**



- Each document has a **topic vector** (a stochastic vector describing the fraction of words that discuss each topic)
- Each topic has a **word vector** (a stochastic vector describing how often a particular word is used in that topic)

Latent Dirichlet Allocation

Topics and documents are **both** described using stochastic vectors:

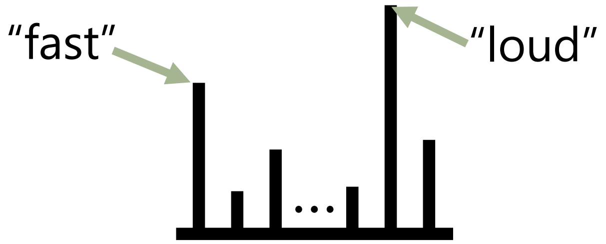


$\theta_{\text{pitch black}}$

Each document has a **topic distribution** which is a mixture over the topics it discusses

number of topics

$\theta_d \in \Delta^K$ i.e., $\forall_d \sum_k \theta_{d,k} = 1$



ϕ_{action}

Each topic has a **word distribution** which is a mixture over the words it discusses

number of words

$\phi_k \in \Delta^D$ i.e., $\forall_k \sum_w \phi_{k,w} = 1$

Latent Dirichlet Allocation

Topics over Time (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models

e.g.

- The topics discussed in conference proceedings progressed from neural networks, towards SVMs and structured prediction (and back to neural networks)
- The topics used in political discourse now cover science and technology more than they did in the 1700s
- With in an institution, e-mails will discuss different topics (e.g. recruiting, conference deadlines) at different times of the year

Latent Dirichlet Allocation

Topics over Time (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models

The ToT model is similar to LDA with one addition:

1. For each topic K, draw a word vector \phi_k from Dir.(\beta)
2. For each document d, draw a topic vector \theta_d from Dir.(\alpha)
3. For each word position i:
    1. draw a topic z_{di} from multinomial \theta_d
    2. draw a word w_{di} from multinomial \phi_{z_{di}}
    3. **draw a timestamp t_{di} from Beta(\psi_{z_{di}})**

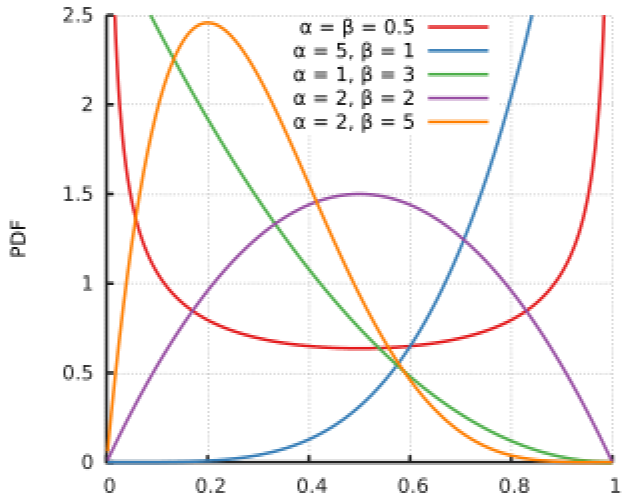
Latent Dirichlet Allocation

Topics over Time (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models

3.3. draw a timestamp t_{di} from Beta(\psi_{z_{di}})

- There is now one Beta distribution **per topic**
- Inference is still done by Gibbs sampling, with an outer loop to update the Beta distribution parameters

Beta distributions are a flexible family of distributions that can capture several types of behavior – e.g. gradual increase, gradual decline, or temporary "bursts"



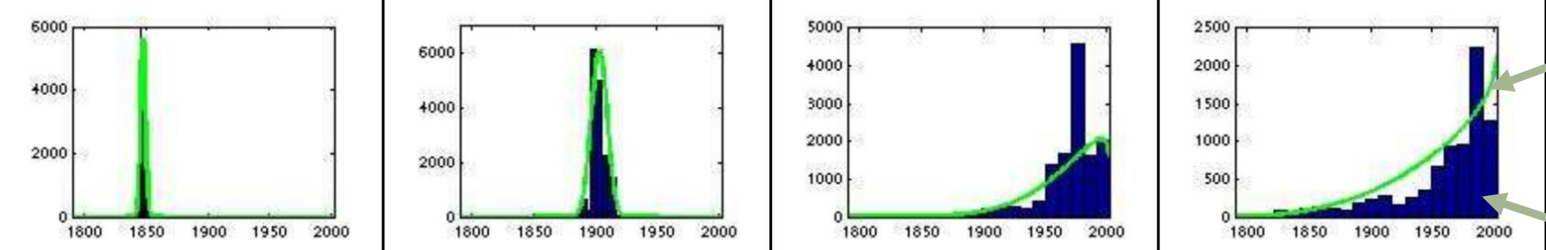
p.d.f.:

$$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$$

Latent Dirichlet Allocation

Results:

Political addresses – the model seems to capture realistic "bursty" and gradually emerging topics



fitted Beta distrbution

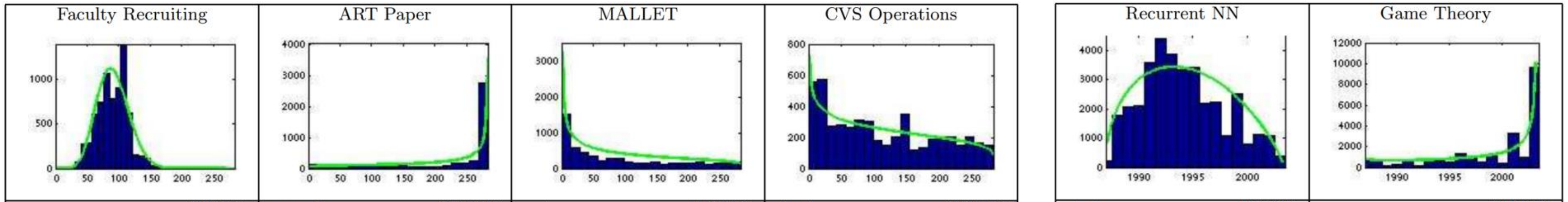
assignments to this topic

Mexican War		Panama Canal		Cold War		Modern Tech	
states	0.02032	government	0.02928	world	0.01875	energy	0.03902
mexico	0.01832	united	0.02132	states	0.01717	national	0.01534
government	0.01670	states	0.02067	security	0.01710	development	0.01448
united	0.01521	islands	0.01167	soviet	0.01664	space	0.01436
war	0.01059	canal	0.01014	united	0.01491	science	0.01227
congress	0.00951	american	0.00872	nuclear	0.01454	technology	0.01227
country	0.00906	cuba	0.00834	peace	0.01408	oil	0.01178
texas	0.00852	made	0.00747	nations	0.01069	make	0.00994
made	0.00727	general	0.00731	international	0.01024	effort	0.00969
great	0.00611	war	0.00660	america	0.00987	administration	0.00957

Latent Dirichlet Allocation

Results:

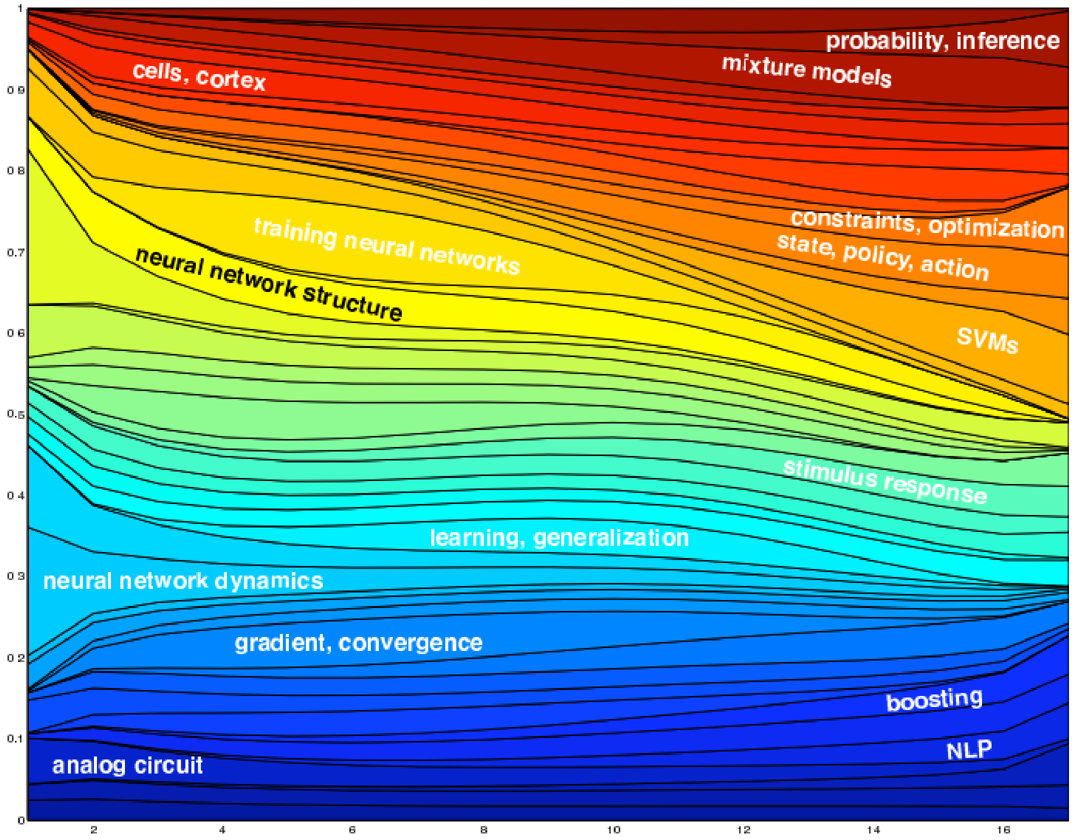
e-mails & conference proceedings



Faculty Recruiting		ART Paper		MALLET		CVS Operations		Recurrent NN		Game Theory	
cs	0.03572	xuerui	0.02113	code	0.05668	check	0.04473	state	0.05963	game	0.02850
april	0.02724	data	0.01814	files	0.04212	page	0.04070	recurrent	0.03765	strategy	0.02378
faculty	0.02341	word	0.01601	mallet	0.04073	version	0.03828	sequence	0.03616	play	0.01490
david	0.02012	research	0.01408	java	0.03085	cvs	0.03587	sequences	0.02462	games	0.01473
lunch	0.01766	topic	0.01366	file	0.02947	add	0.03083	time	0.02402	player	0.01451
schedule	0.01656	model	0.01238	al	0.02479	update	0.02539	states	0.02057	agents	0.01346
candidate	0.01560	andres	0.01238	directory	0.02080	latest	0.02519	transition	0.01300	expert	0.01281
talk	0.01355	sample	0.01152	version	0.01664	updated	0.02317	finite	0.01242	strategies	0.01123
bruce	0.01273	enron	0.01067	pdf	0.01421	checked	0.02277	length	0.01154	opponent...	0.01088
visit	0.01232	dataset	0.00960	bug	0.01352	change	0.02156	strings	0.01013	nash	0.00848

Latent Dirichlet Allocation

Results:
conference proceedings (NIPS)



Relative weights of various topics in 17 years of NIPS proceedings

Learning Outcomes

- Discussed how text can change over time

References

Further reading:
"Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends"
(Wang & McCallum, 2006)
http://people.cs.umass.edu/~mccallum/papers/tot-kdd06.pdf

Web Mining and Recommender Systems

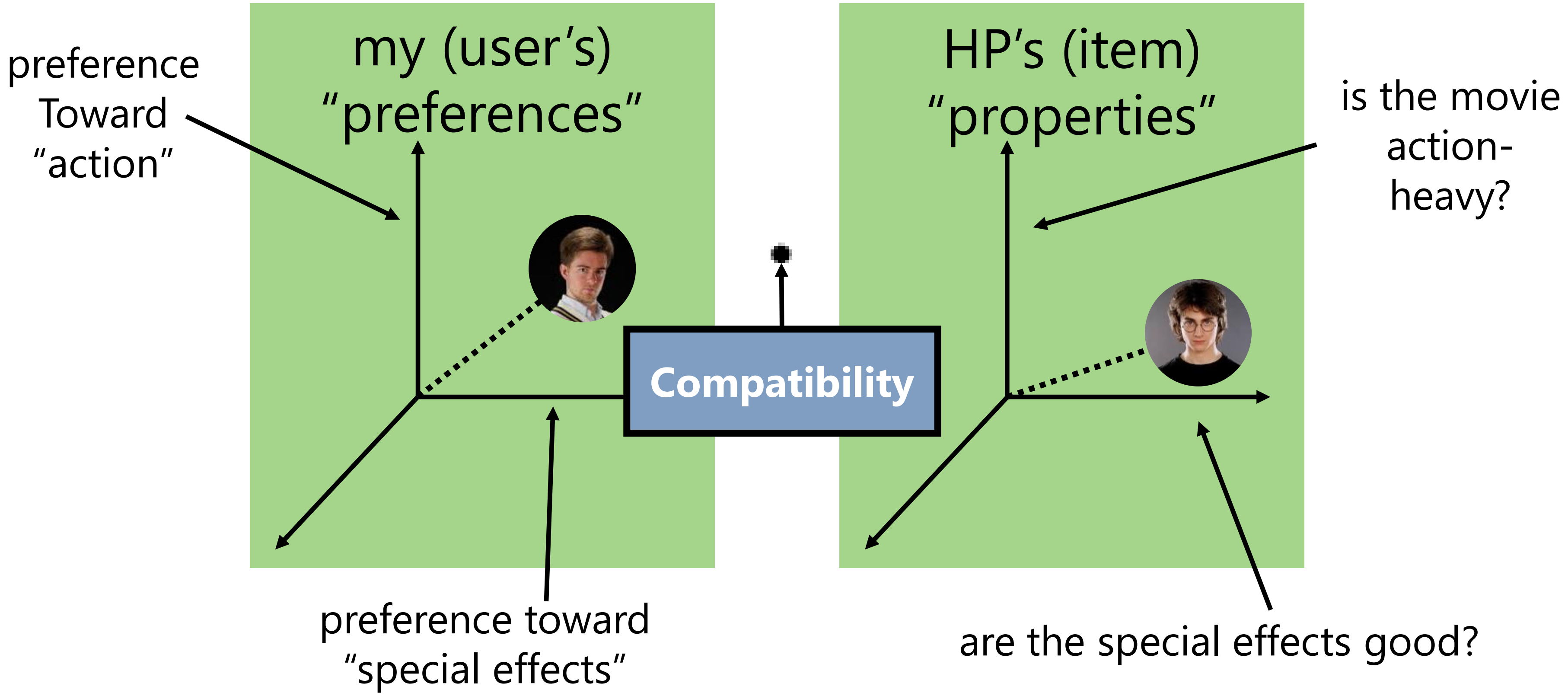
Temporal recommender systems

Learning Goals

- Discuss how temporal dynamics can be incorporated into recommender systems

Previously...

Recommender Systems go beyond the methods we've seen so far by trying to model the **relationships** between people and the items they're evaluating



Previously...

Predict a user's rating of an item according to:

$$f(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

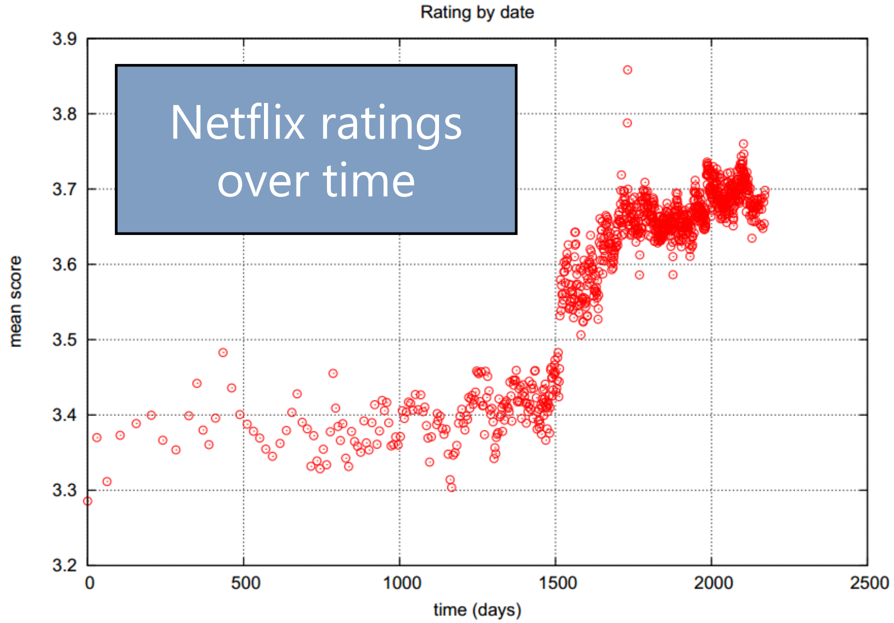
By solving the optimization problem:

$$\arg\min_{\alpha,\beta,\gamma} \underbrace{\sum_{u,i}(\alpha+\beta_u+\beta_i+\gamma_u\cdot\gamma_i-R_{u,i})^2}_{\text{error}}+\lambda \underbrace{\left[\sum_u \beta_u^2 + \sum_i \beta_i^2 + \sum_i \|\gamma_i\|_2^2 + \sum_u \|\gamma_u\|_2^2\right]}_{\text{regularizer}}$$

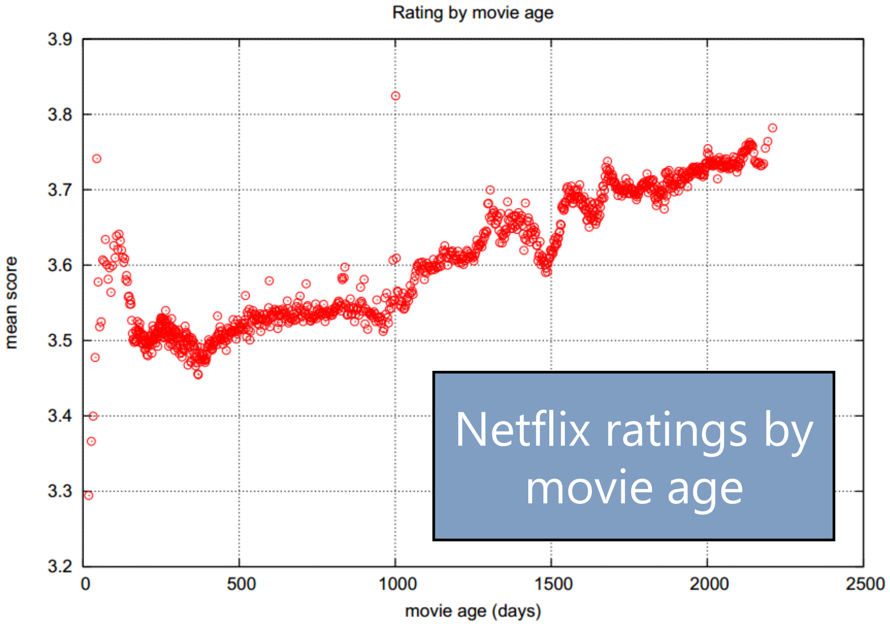
(e.g. using stochastic gradient descent)

Temporal latent-factor models

To build a reliable system (and to win the Netflix prize!) we need to account for **temporal dynamics:**



(Netflix changed their interface)



(People tend to give higher ratings to older movies)

So how was this actually done?

Figure from Koren: "Collaborative Filtering with Temporal Dynamics" (KDD 2009)

Temporal latent-factor models

To start with, let's just assume that it's only the **bias** terms that explain these types of temporal variation (which, for the examples on the previous slides, is potentially enough)

$$b_{u,i}(t) = \alpha + \beta_u(t) + \beta_i(t)$$

Idea: temporal dynamics for *items* can be explained by long-term, gradual changes, whereas for users we'll need a different model that allows for "bursty", short-lived behavior

Temporal latent-factor models

temporal bias model:

$$b_{u,i}(t) = \alpha + \beta_u(t) + \beta_i(t)$$

For item terms, just separate the dataset into (equally sized) bins:*

$$\beta_i(t) = \beta_i + \beta_{i,\text{Bin}(t)}$$

*in Koren's paper they suggested ~30 bins corresponding to about 10 weeks each for Netflix

or bins for periodic effects (e.g. the day of the week):

$$\beta_i(t) = \beta_i + \beta_{i,\text{Bin}(t)} + \beta_{i,\text{period}(t)}$$

What about user terms?

- We need something much finer-grained
- **But** – for most users we have far too little data to fit very short term dynamics

Temporal latent-factor models

Start with a simple model of drifting dynamics for users:

mean rating
date for user u

hyperparameter
(ended up as x=0.4 for Koren)

$$\mathrm{dev}_u(t) = \underbrace{\mathrm{sign}(t - t_u)}_{\text{before (-1) or after (1) the mean date}} \cdot |\underbrace{t - t_u}_{\text{days away from mean date}}|^x$$

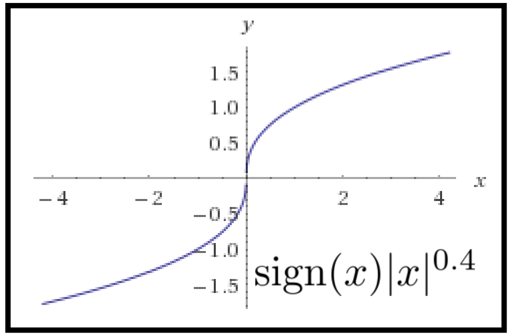
Temporal latent-factor models

Start with a simple model of drifting dynamics for users:

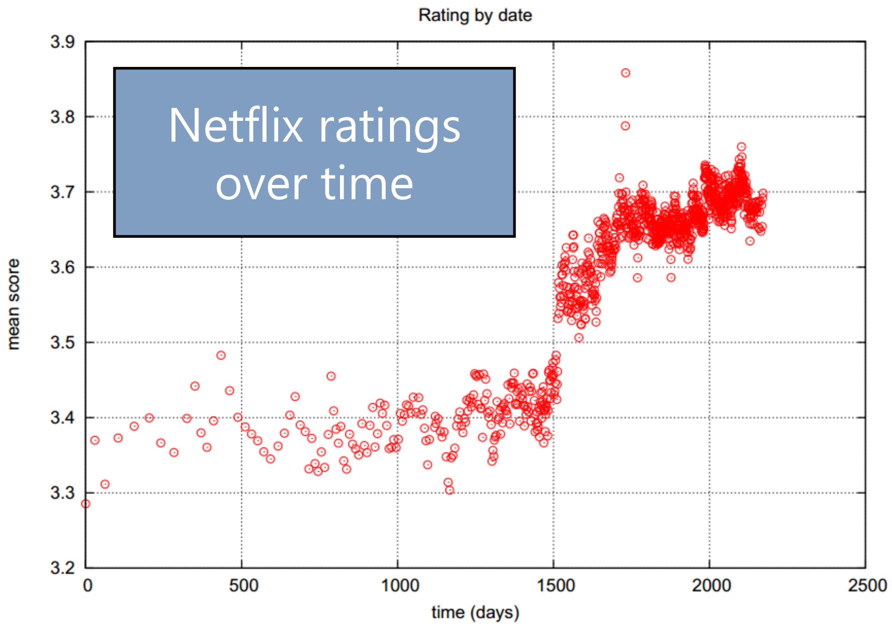
$$\mathrm{dev}_u(t) = \underbrace{\mathrm{sign}(t - \overset{\text{mean rating date for user u}}{t_u})}_{\text{before (-1) or after (1) the mean date}} \cdot \underbrace{|t - t_u|}_{\text{days away from mean date}}^{\overset{\text{hyperparameter (ended up as x=0.4 for Koren)}}{x}}$$

time-dependent user bias can then be defined as:

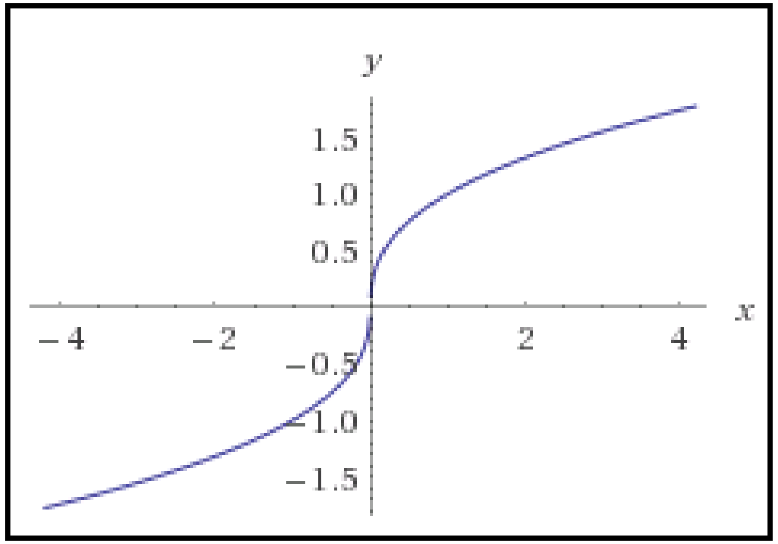
$$\beta_u^{(1)}(t) = \underset{\text{overall user bias}}{\beta_u} + \underset{\text{sign and scale for deviation term}}{\alpha_u} \cdot \mathrm{dev}_u(t)$$



Temporal latent-factor models



Real data



Fitted model

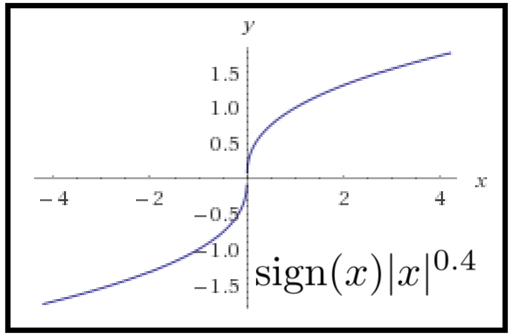
Temporal latent-factor models

time-dependent user bias can then be defined as:

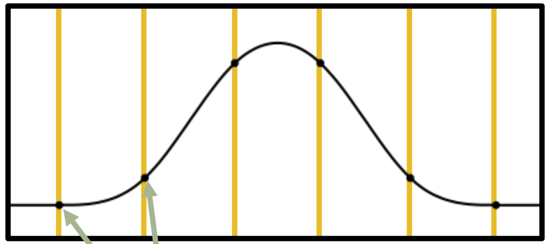
$$\beta_u^{(1)}(t) = \beta_u + \alpha_u \cdot \mathrm{dev}_u(t)$$

overall user bias

sign and scale for deviation term



- Requires only two parameters per user and captures some notion of temporal "drift" (even if the model found through cross-validation is (to me) completely unintuitive)
- To develop a slightly more expressive model, we can interpolate smoothly between biases using splines



Temporal latent-factor models

number of control
points for this user
(k_u = n_u^0.25 in Koren)

user bias associated
with this control point

$$\beta_u^{(2)}(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}}$$

time associated
with control point
(uniformly spaced)

Temporal latent-factor models

number of control points for this user
(k_u = n_u^0.25 in Koren)

user bias associated with this control point

$$\beta_u^{(2)}(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}}$$

time associated with control point
(uniformly spaced)

- This is now a reasonably flexible model, but still only captures *gradual drift*, i.e., it can't handle sudden changes (e.g. a user simply having a bad day)

Temporal latent-factor models

- Koren got around this just by adding a "per-day" user bias:

$$\underbrace{\beta_{u,t}}_{\text{bias for a particular day (or session)}}$$

- Of course, this is only useful for particular days in which users have a lot of (abnormal) activity
- The final (time-evolving bias) model then combines all of these factors:

$$\beta_{u,i}(t) = \overbrace{\alpha}^{\text{global offset}} + \underbrace{\beta_u}_{\text{user bias}} + \overbrace{\alpha_u \cdot \mathrm{dev}_u(t)}^{\text{gradual deviation (or splines)}} + \underbrace{\beta_{u,t}}_{\text{single-day dynamics}} + \overbrace{\beta_i}^{\text{item bias}} + \overbrace{\beta_{i,\mathrm{Bin}(t)}}^{\text{gradual item bias drift}}$$

Temporal latent-factor models

Finally, we can add a time-dependent scaling factor:

$$\beta_{u,i}(t) = \alpha + \beta_u + \alpha_u \cdot \mathrm{dev}_u(t) + \beta_{u,t} + (\beta_i + \beta_{i,\mathrm{Bin}(t)}) \cdot c_u(t)$$

also defined as $c_u + c_{u,t}$

Latent factors can also be defined to evolve in the same way:

$$\gamma_{u,k}(t) = \gamma_{u,k} + \alpha_{u,k} \cdot \mathrm{dev}_u(t) + \gamma_{u,k,t}$$

factor-dependent
user drift

factor-dependent
short-term effects

Summary

- Effective modeling of temporal factors was absolutely critical to this solution outperforming alternatives on Netflix's data
  - In fact, even with only temporally evolving *bias* terms, their solution was already ahead of Netflix's previous ("Cinematch") model

On the other hand…

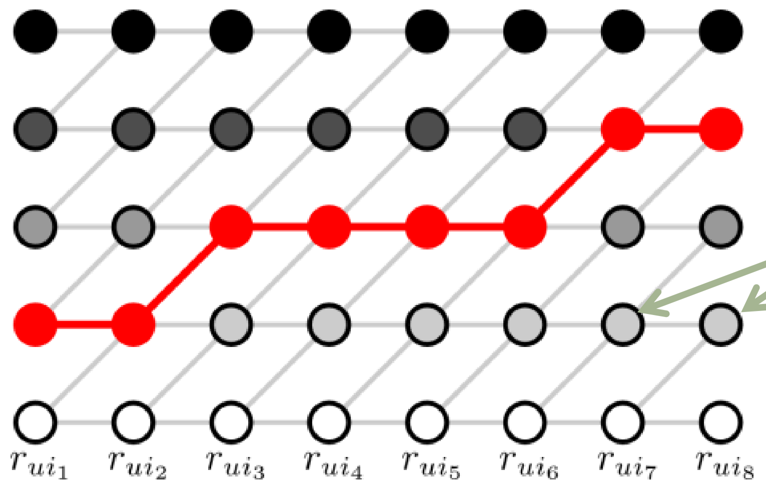
- Many of the ideas here depend on dynamics that are quite specific to "Netflix-like" settings
- Some factors (e.g. short-term effects) depend on a high density of data per-user and per-item, which is not always available

Temporal latent-factor models

Summary

- Changing the setting, e.g. to model the stages of progression through the symptoms of a disease, or even to model the temporal progression of people's opinions on beers, means that alternate temporal models are required

rows: models of increasingly "experienced" users



columns: review timeline for one user

Learning Outcomes

- Discussed how temporal dynamics can be incorporated into recommender systems
- Discussed how this was useful for Netflix in particular

References

Further reading:
"Collaborative filtering with temporal dynamics"
Yehuda Koren, 2009
http://research.yahoo.com/files/kdd-fp074-koren.pdf