

# Web Mining and Recommender Systems

Temporal data mining

## Temporal models

This week we'll 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

# Web Mining and Recommender Systems

Regression for sequence data

# Week 1 – Regression

Given **labeled training data** of the form

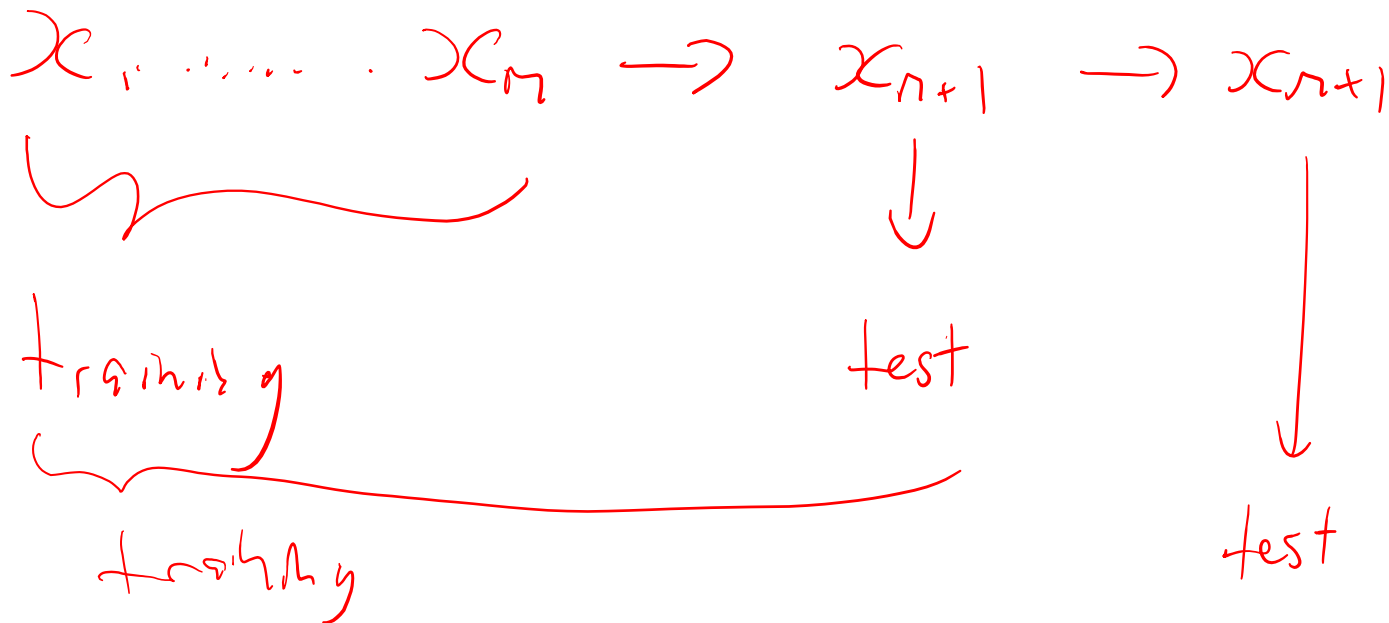
$$\{(\text{data}_1, \text{label}_1), \dots, (\text{data}_n, \text{label}_n)\}$$

Infer the function

$$f(\text{data}) \xrightarrow{?} \text{labels}$$

# Time-series regression

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



# Time-series regression

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

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

$K \leftarrow$  window size


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

$$O(m \times K)$$

# Time-series regression

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

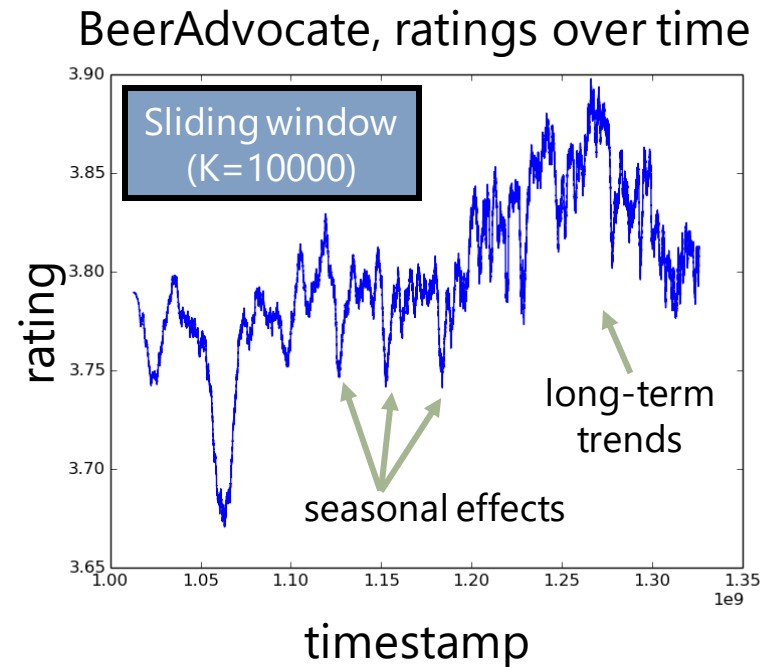
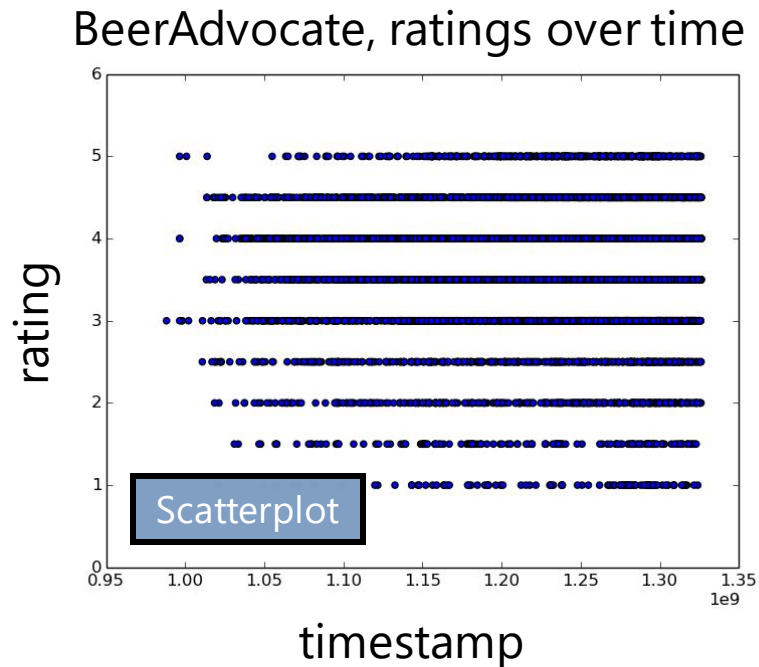
- This can be computed efficiently via dynamic programming:

$$f(x_1, \dots, x_{m+1}) = \frac{K f(x_1 + \dots + x_n) + x_{n-K+1} + x_{n+1}}{K}$$


$O(n+K)$  vs.  $O(n \times K)$

# Time-series regression

Also useful to plot data:



Code on:

<http://jmcauley.ucsd.edu/code/week10.py>



# Time-series regression

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

$$f(x_1, \dots, x_m) = \frac{Kx_m + (K-1)x_{m-1} + \dots + 1x_{m-K+1}}{1 + 2 + \dots + K}$$

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



# Time-series regression

**Method 3:** weight the most recent points exponentially higher

$$f(x_1) = x_1$$

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

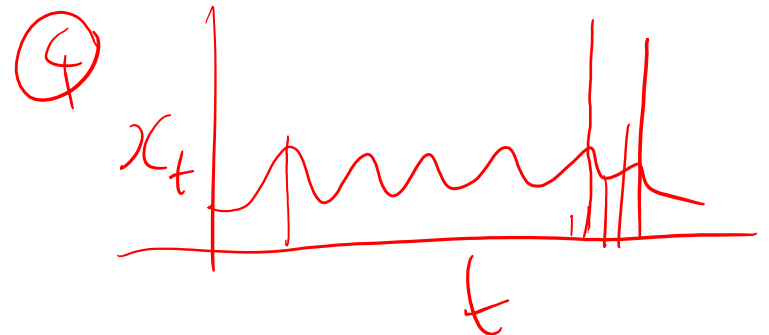
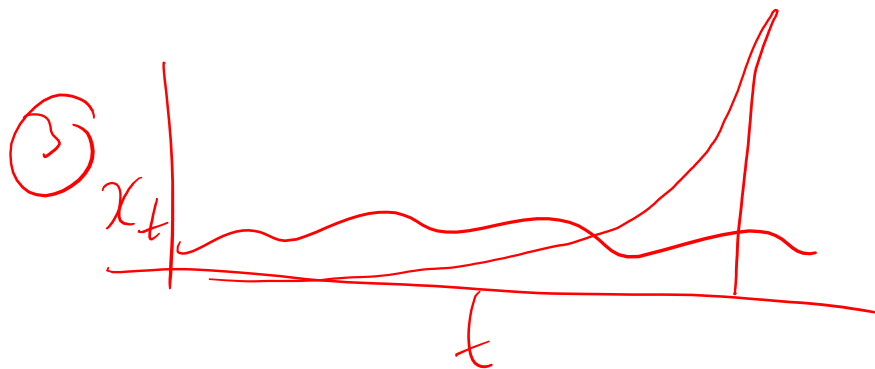
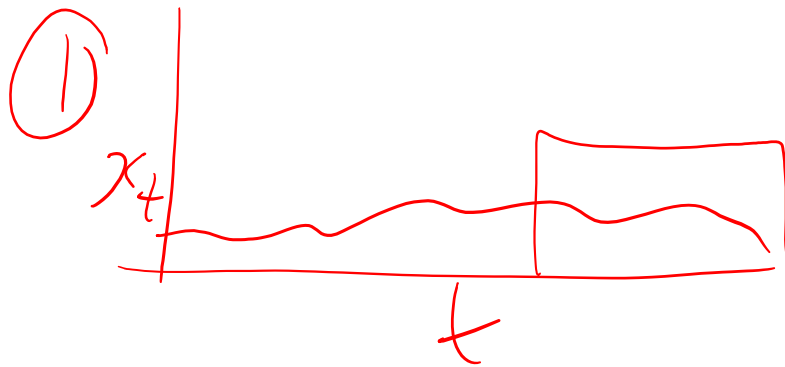


# Methods 1, 2, 3

Method 1: Sliding window

Method 2: Linear decay

Method 3: Exponential decay



# 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?

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

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

$$\theta = \underset{\theta}{\text{argmin}} \sum_n \left( f(x_1, \dots, x_n) - x_{n+1} \right)^2$$

# 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

# Web Mining and Recommender Systems

Temporal dynamics of social networks

## 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

Two weeks ago 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.

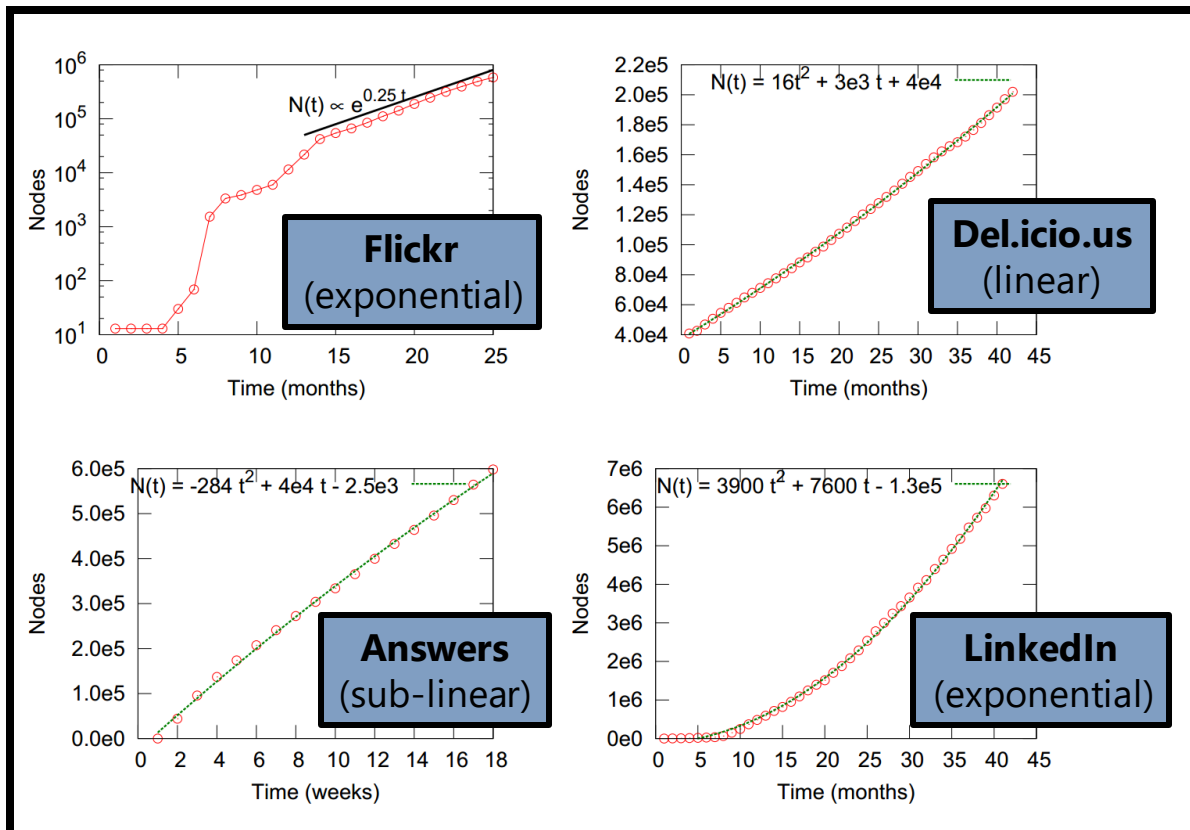
But if we observe the **order** in which edges were created, we can study how these phenomena change as a function of time

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?



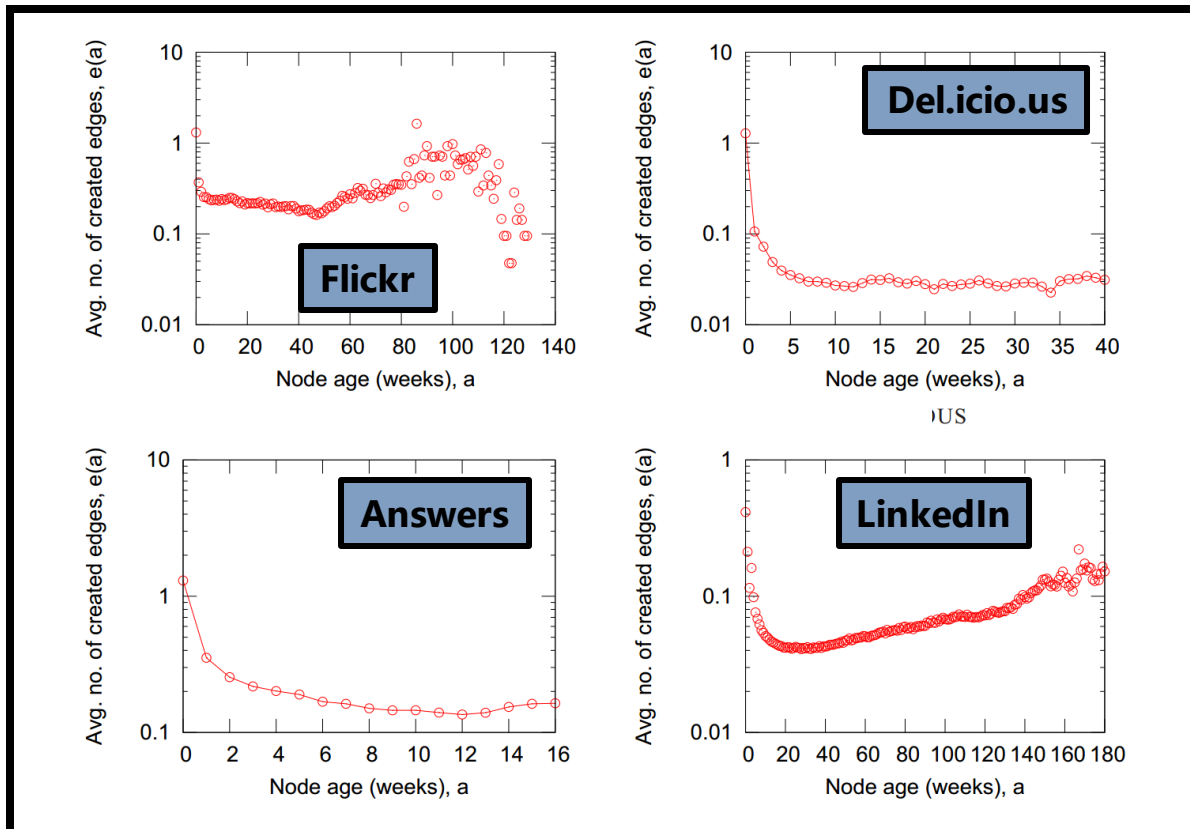
(from Leskovec,  
2008 (CMU Thesis))

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

# 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

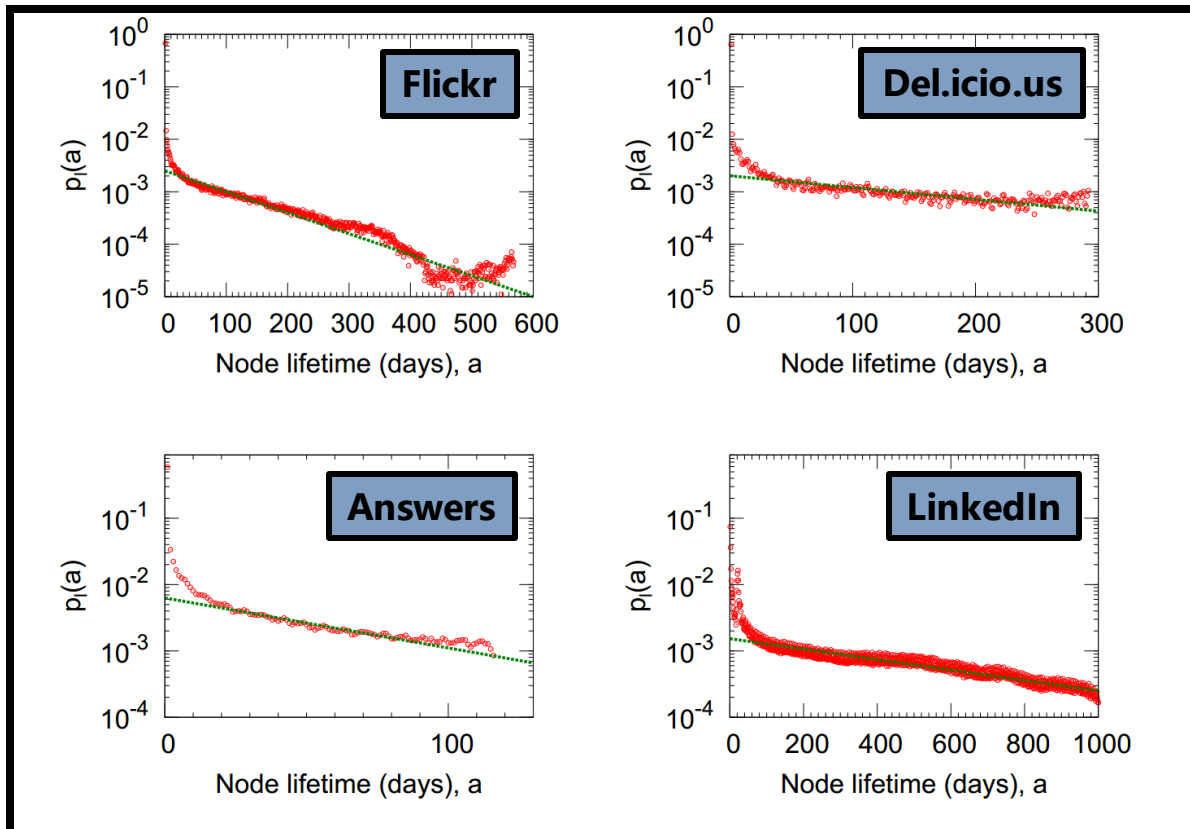


**A:** In most networks there's a "burst" of initial edge creation which gradually flattens out. Very different behavior on LinkedIn (guesses as to why?)

# 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



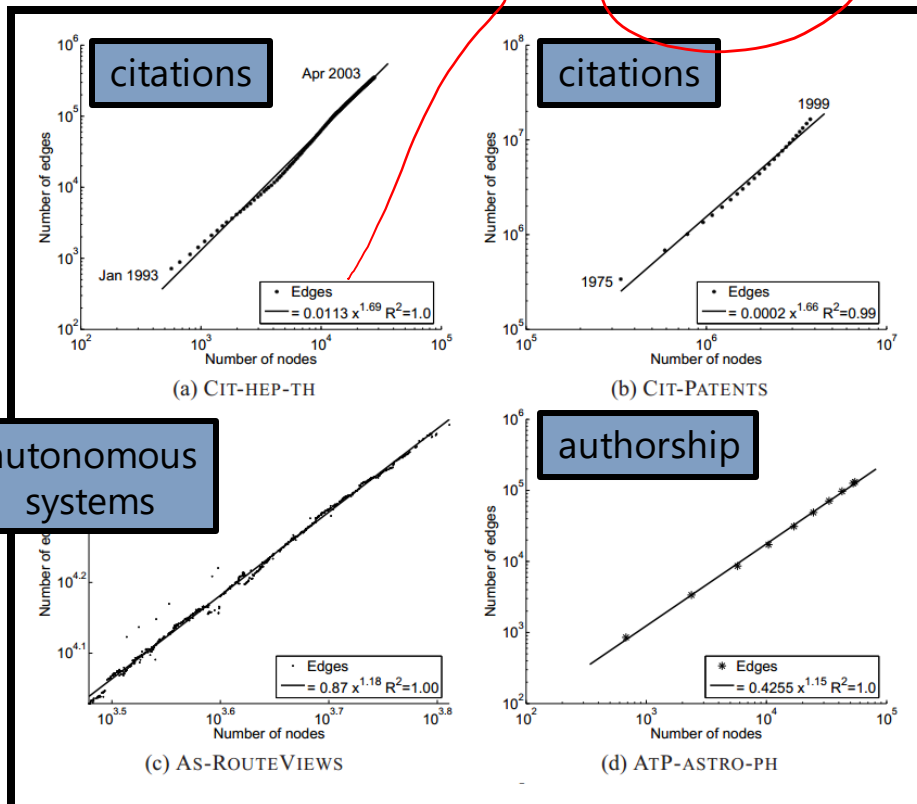
**A:** Node lifetimes follow a power-law: many many nodes are shortlived, with a long-tail of older nodes

# 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?

$0.0113 x^{1.69}$



- 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

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

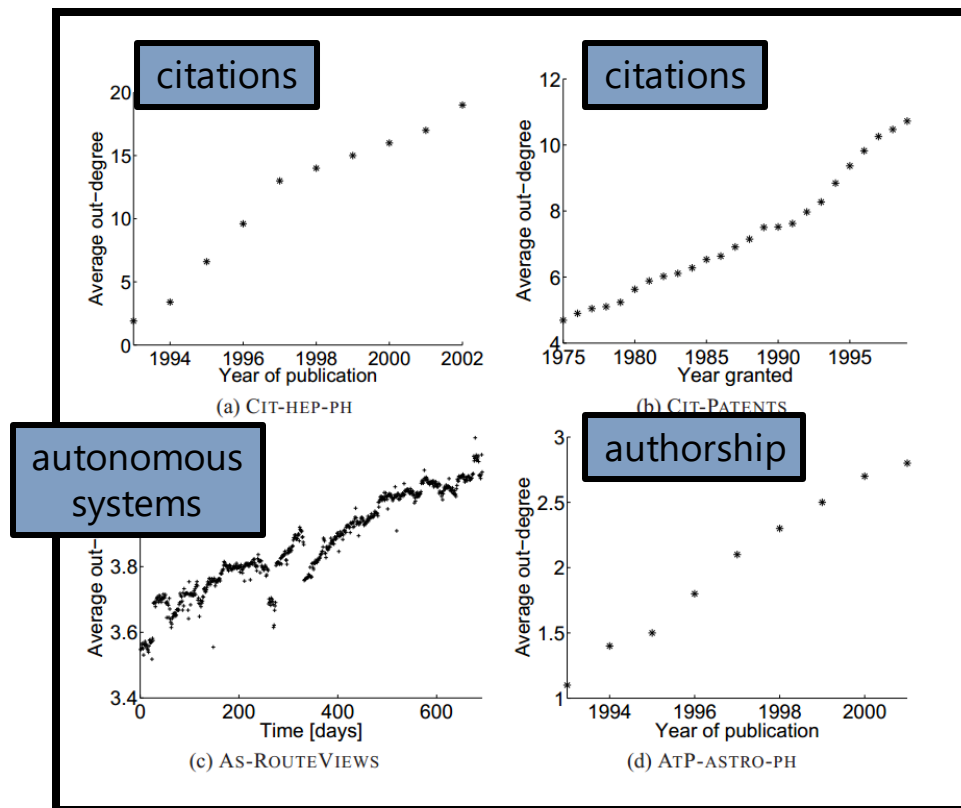
where

$$1 \leq a \leq 2$$

- $a = 1$  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

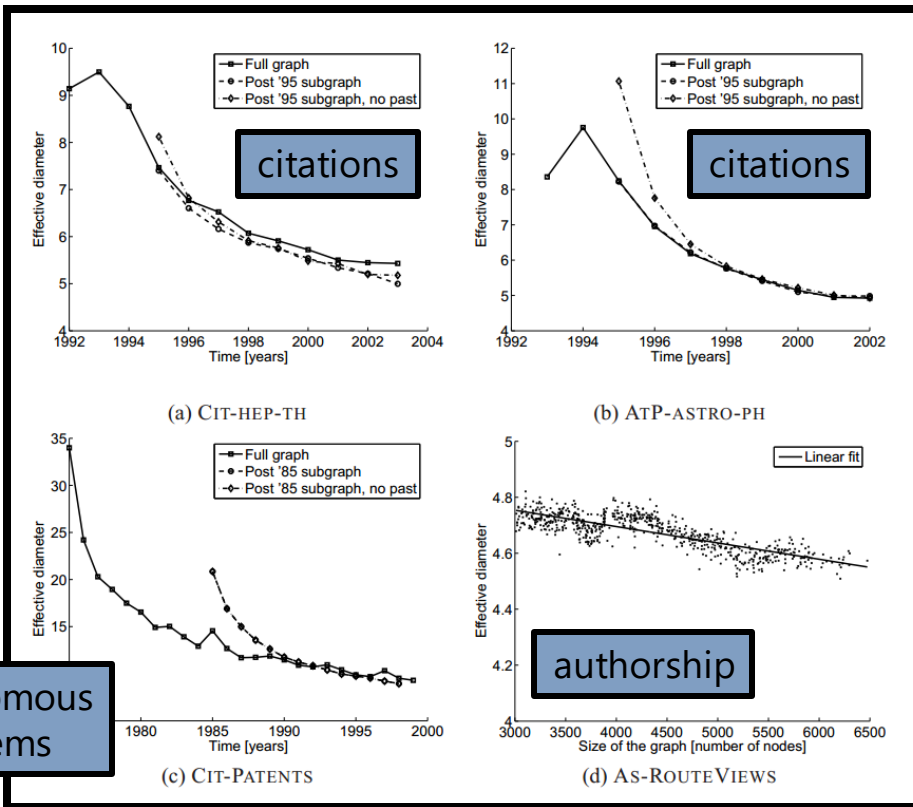
**Q2:** How does the degree change over time?



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

# Temporal dynamics of social networks

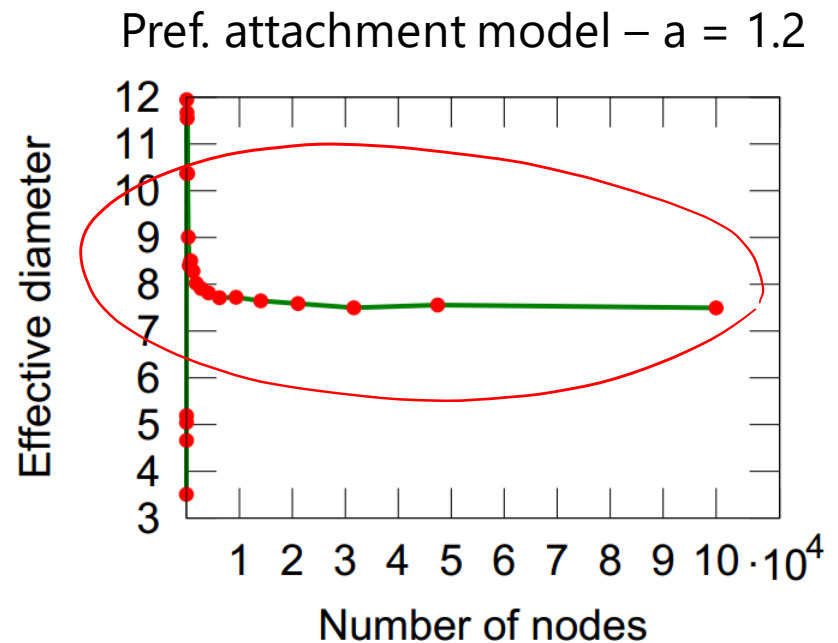
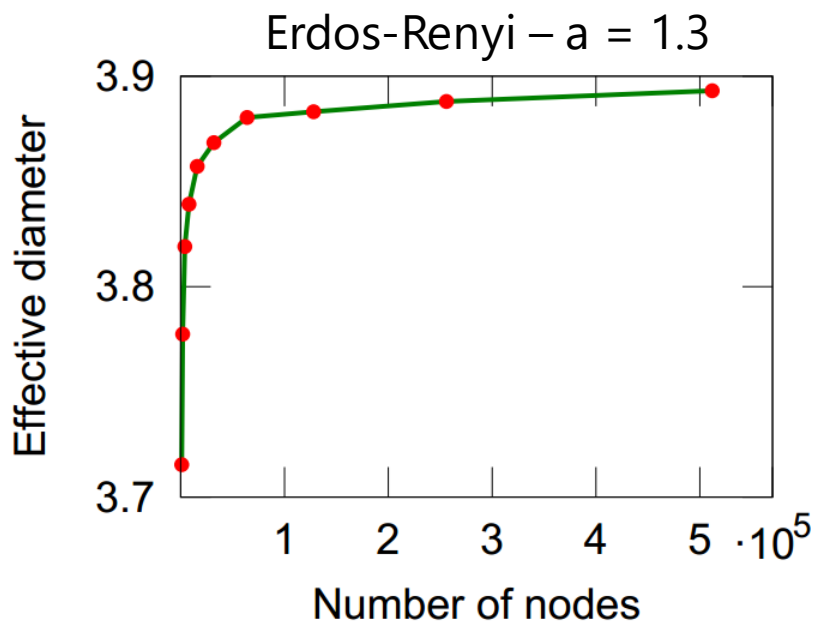
**Q3:** If the network becomes **denser**, what happens to the (effective) diameter?



- **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:



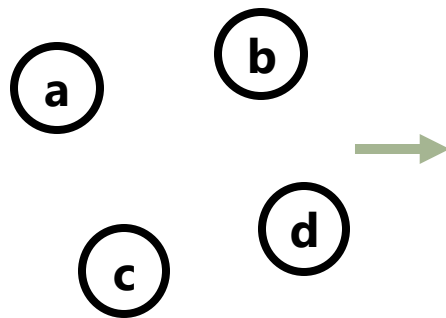


# Temporal dynamics of social networks

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

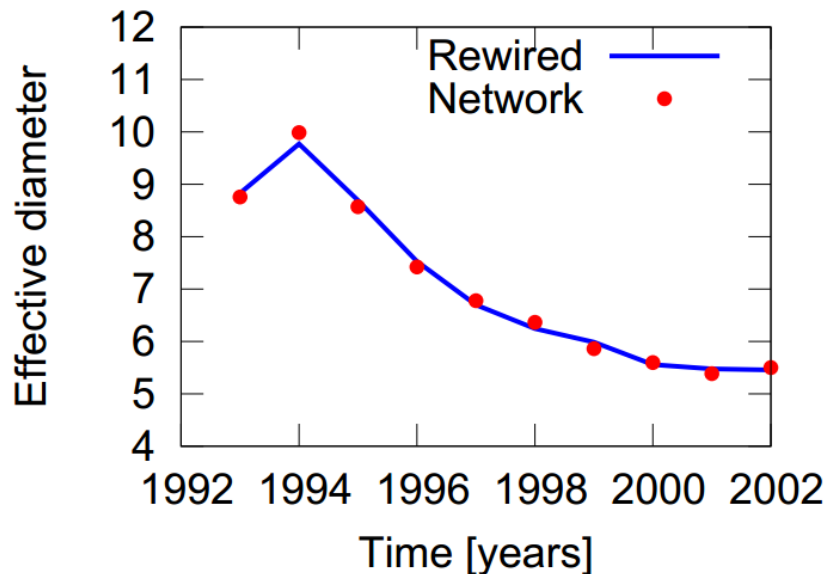


random rewiring preserves the degree distribution, and randomly samples amongst networks with observed degree distribution

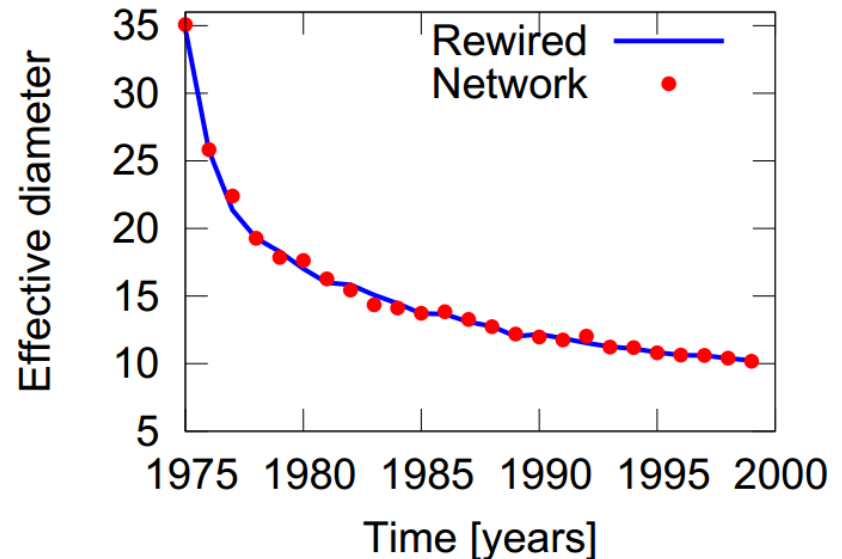
# Temporal dynamics of social networks

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

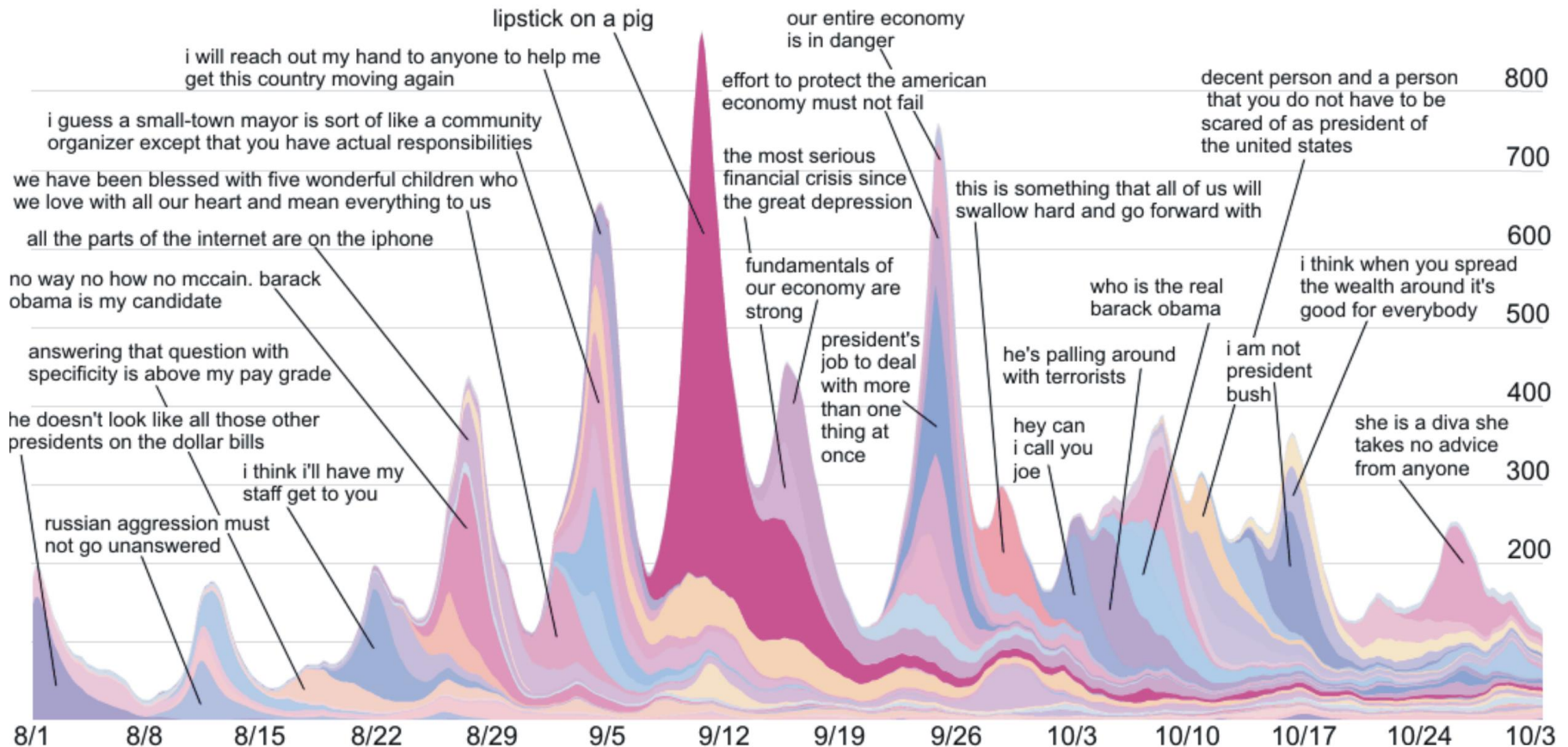
So, a decreasing diameter is **not** a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model

**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

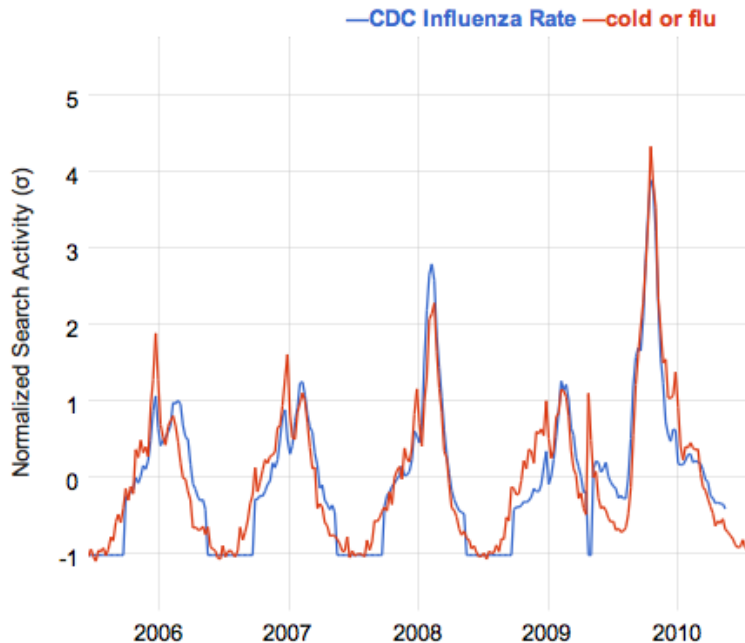
## Other interesting topics...



"memetracker"

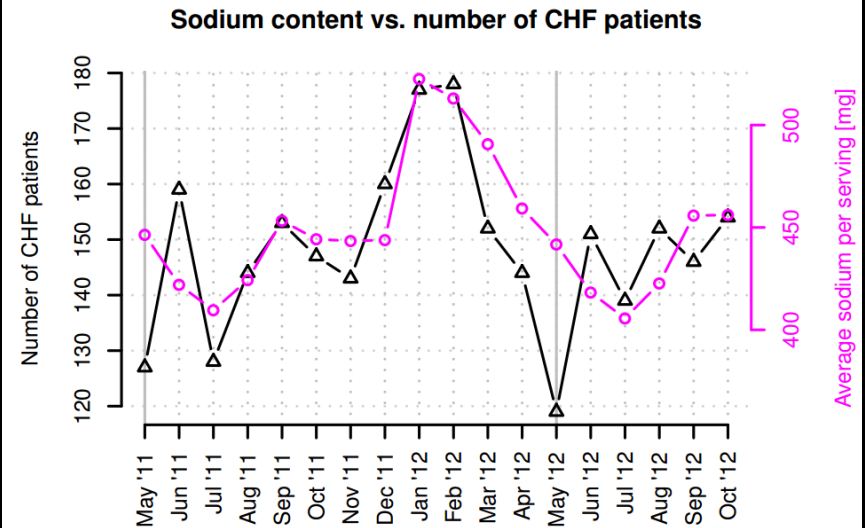
# Temporal dynamics of social networks

## 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](http://infolab.stanford.edu/~west1/pubs/West-White-Horvitz_WWW-13.pdf)

# Questions?

## 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

# 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.*



$F_{\text{text}} = [150, 0, 0, 0, 0, 0, \dots, 0]$

a

aardvark

zoetrope

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





# Latent Dirichlet Allocation

In week 5, 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

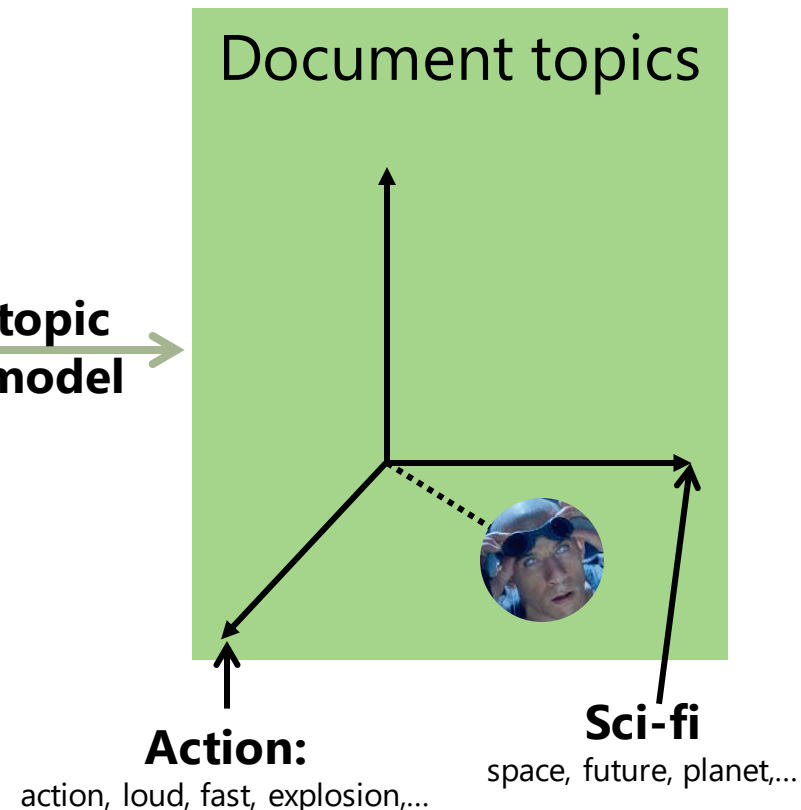
This review is from: [The Chronicles of Riddick \(Widescreen Unrated Director's Cut\) \(DVD\)](#)

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.

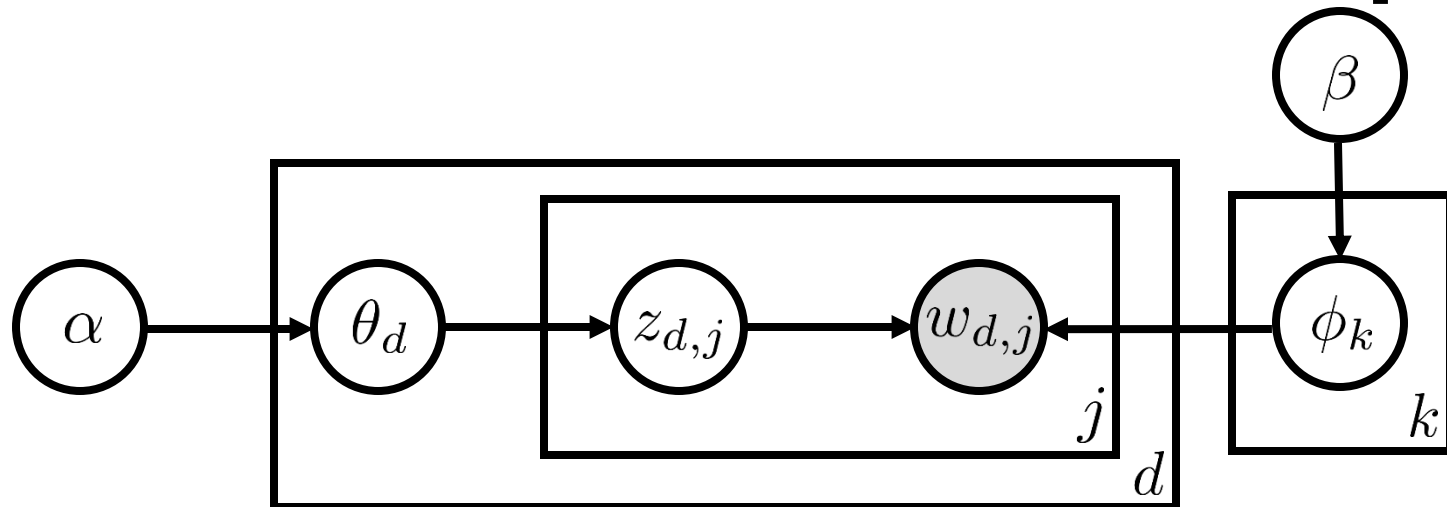
(review of "The Chronicles of Riddick")

topic  
model →



# Latent Dirichlet Allocation

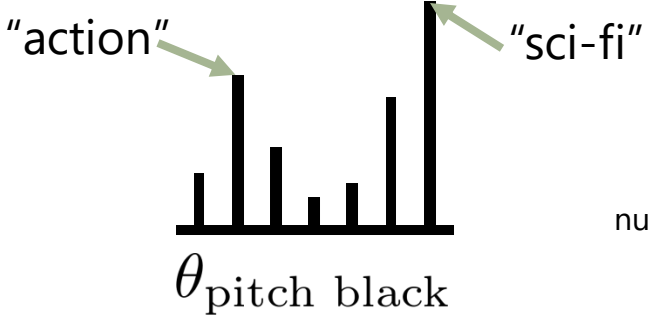
We saw how **LDA** can be used to describe documents in terms of **topics**



- 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:




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

number of topics  $\rightarrow$

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

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



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

number of words  $\rightarrow$

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

$\phi_{\text{action}}$

# 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
- Within 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  $\text{Dir}(\beta)$
2. For each document  $d$ , draw a topic vector  $\theta_d$  from  $\text{Dir}(\alpha)$
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  $\text{Beta}(\psi_{\{z_{\{di\}}\}})$**

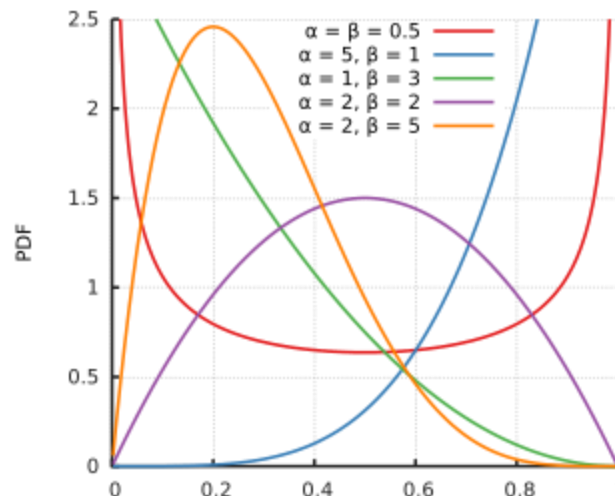
# Latent Dirichlet Allocation

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

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

- 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”



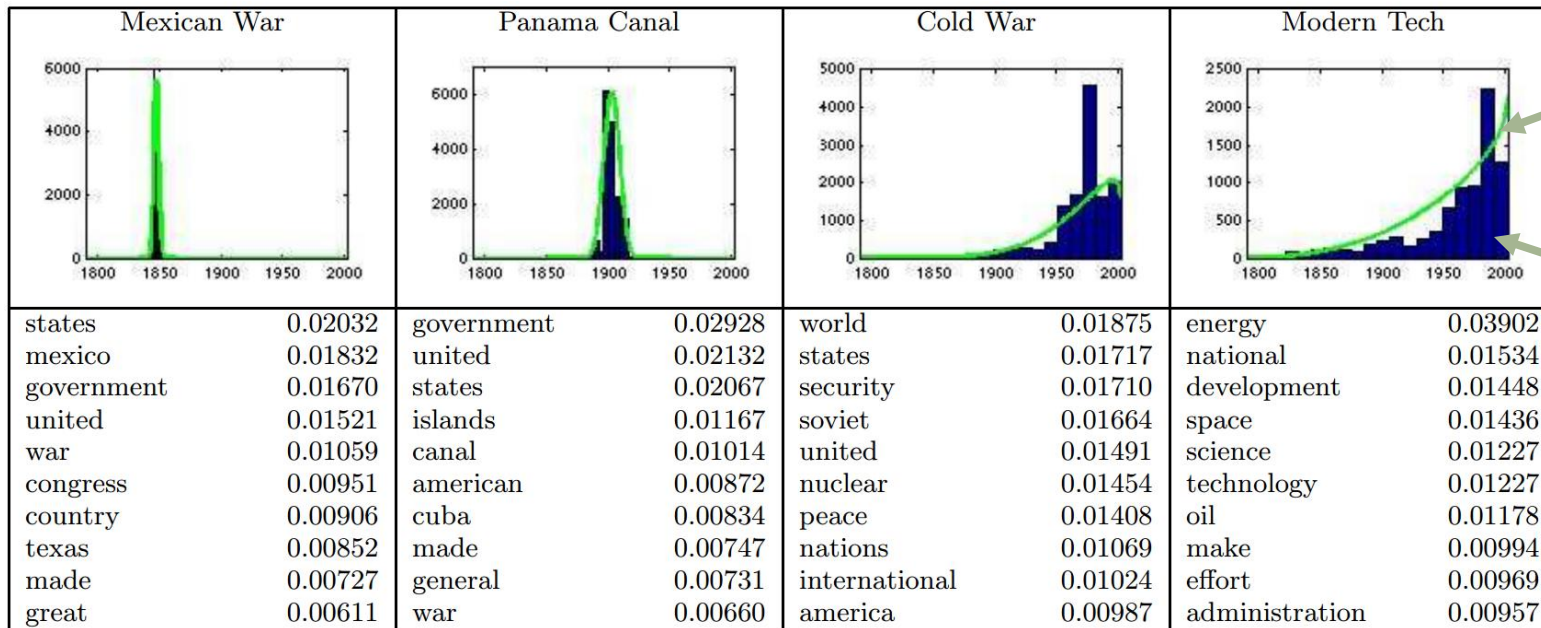
p.d.f.:

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

# Latent Dirichlet Allocation

## Results:

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

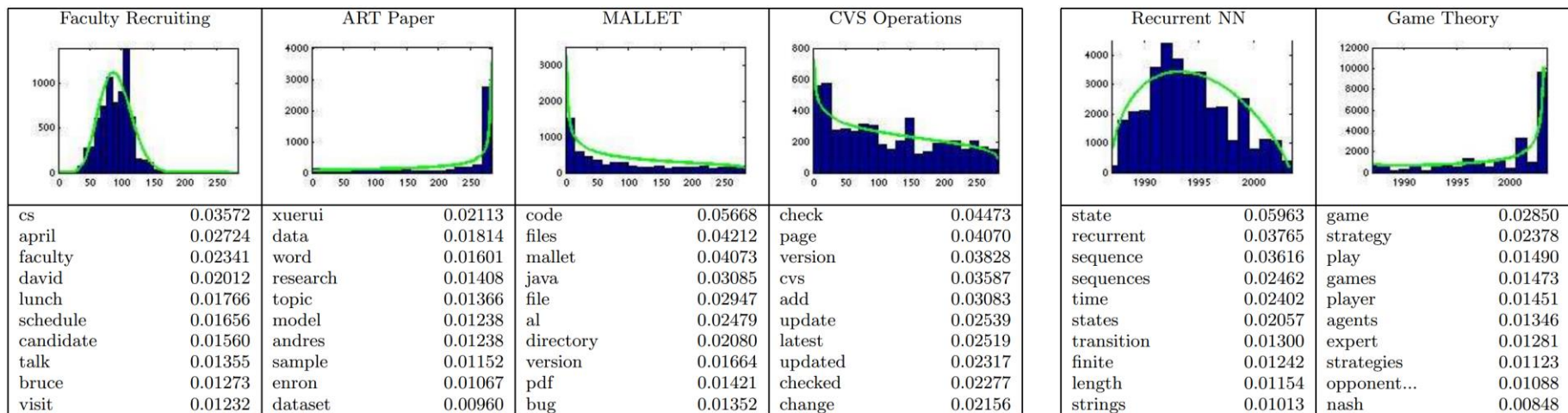


fitted Beta distribution

assignments to this topic

# Latent Dirichlet Allocation

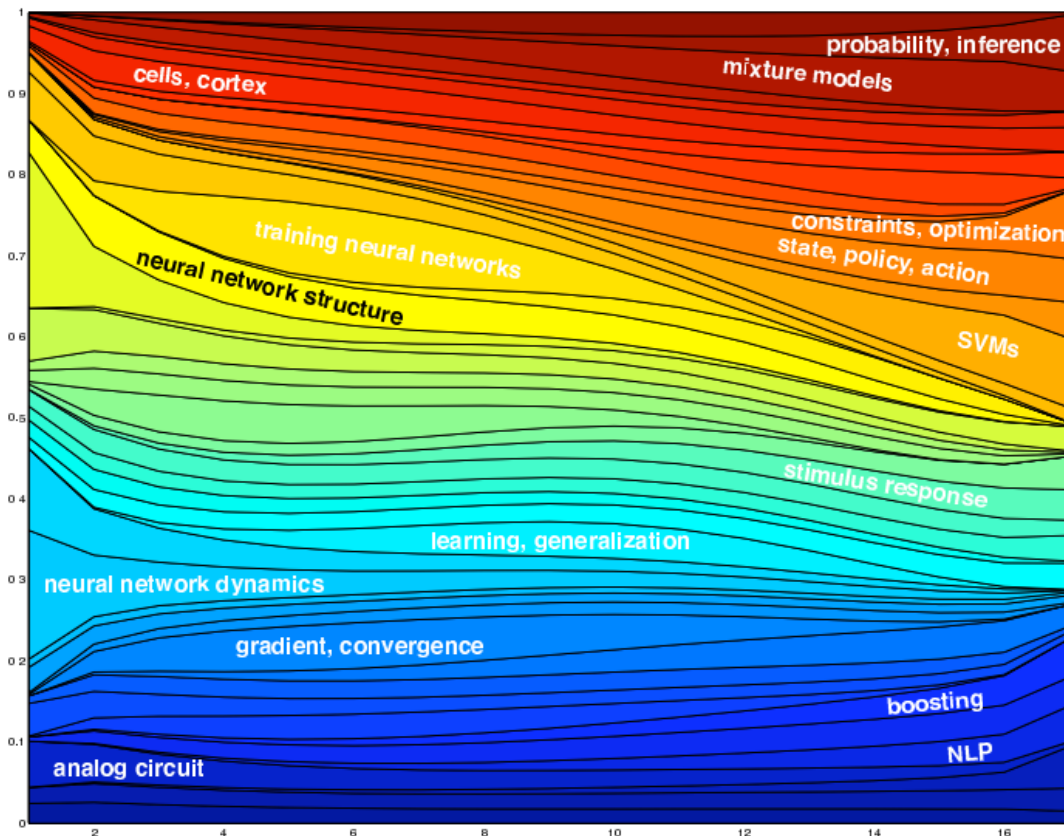
## Results: e-mails & conference proceedings





# Latent Dirichlet Allocation

**Results:**  
conference proceedings (NIPS)



Relative weights  
of various topics  
in 17 years of  
NIPS proceedings

# Questions?

Further reading:

“Topics over Time: A Non-Markov  
Continuous-Time Model of Topical  
Trends”

(Wang & McCallum, 2006)

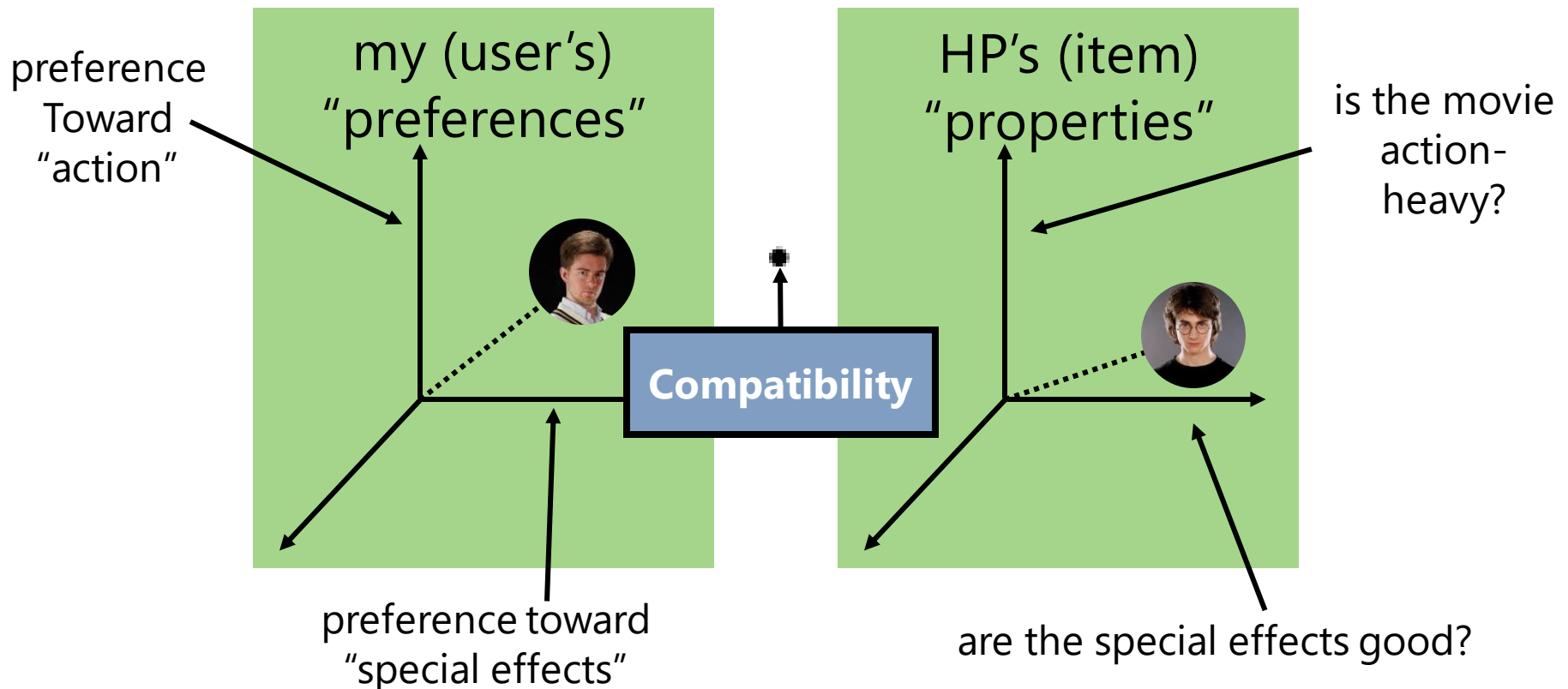
<http://people.cs.umass.edu/~mccallum/papers/tot-kdd06.pdf>

# Web Mining and Recommender Systems

Temporal recommender systems

# Week 4

**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



# Week 4

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

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

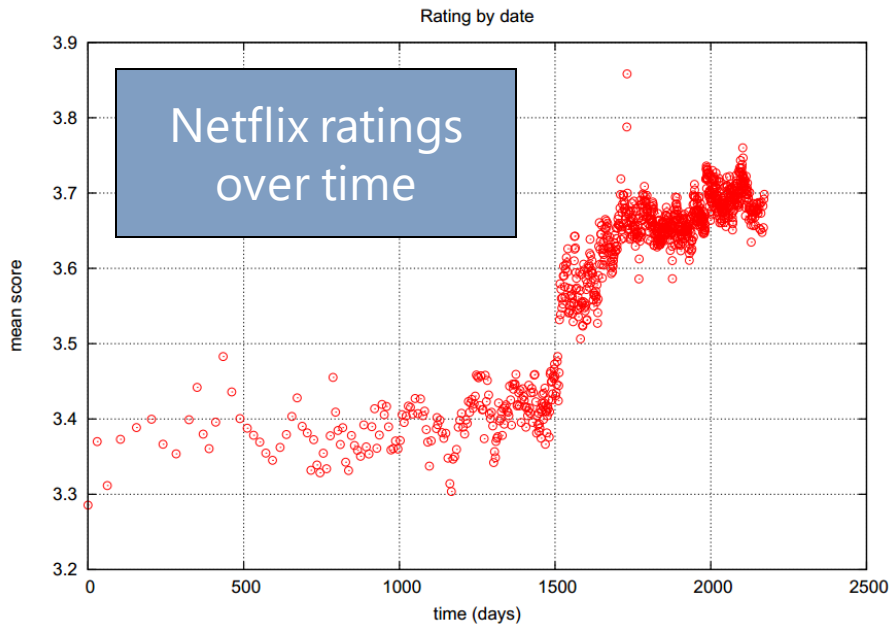
By solving the optimization problem:

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

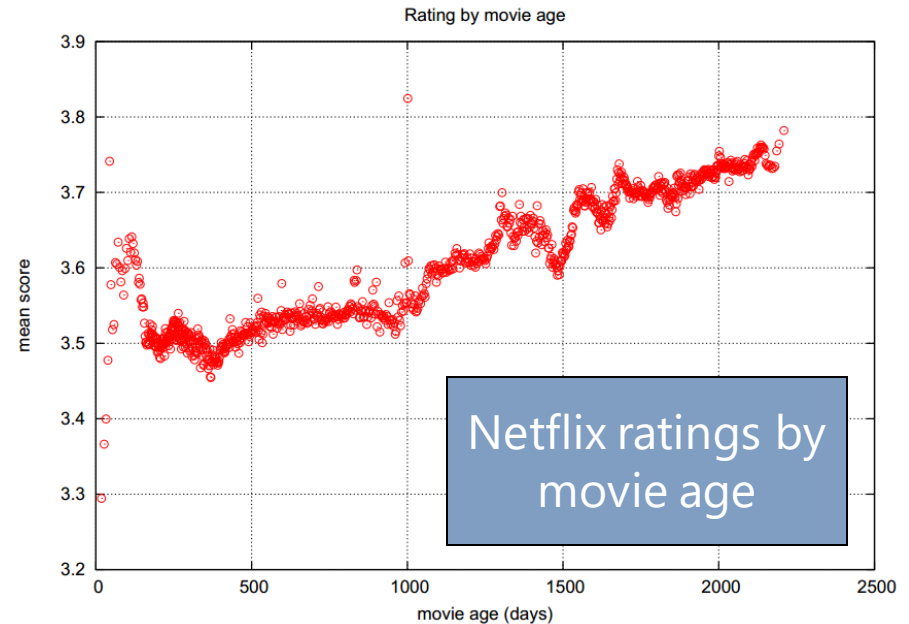
(e.g. using stochastic gradient descent)

# 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?

# 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:

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

**mean** rating  
date for user  $u$

hyperparameter  
(ended up as  $x=0.4$  for Koren)

The diagram shows the equation  $\text{dev}_u(t) = \text{sign}(t - t_u) \cdot |t - t_u|^x$ . A green arrow points from the text 'mean rating date for user u' to the variable  $t_u$ . Another green arrow points from the text 'hyperparameter (ended up as x=0.4 for Koren)' to the exponent  $x$ . A green bracket under the  $\text{sign}(t - t_u)$  term is labeled 'before (-1) or after (1) the mean date'. A second green bracket under the  $|t - t_u|^x$  term is labeled 'days away from mean date'.

# Temporal latent-factor models

Start with a simple model of drifting dynamics for users:

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

**mean** rating  
date for user  $u$

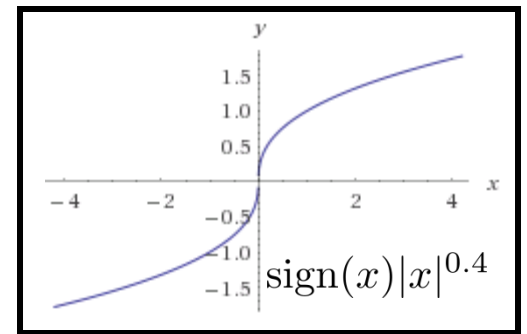
hyperparameter  
(ended up as  $x=0.4$  for Koren)

time-dependent user bias can then be defined as:

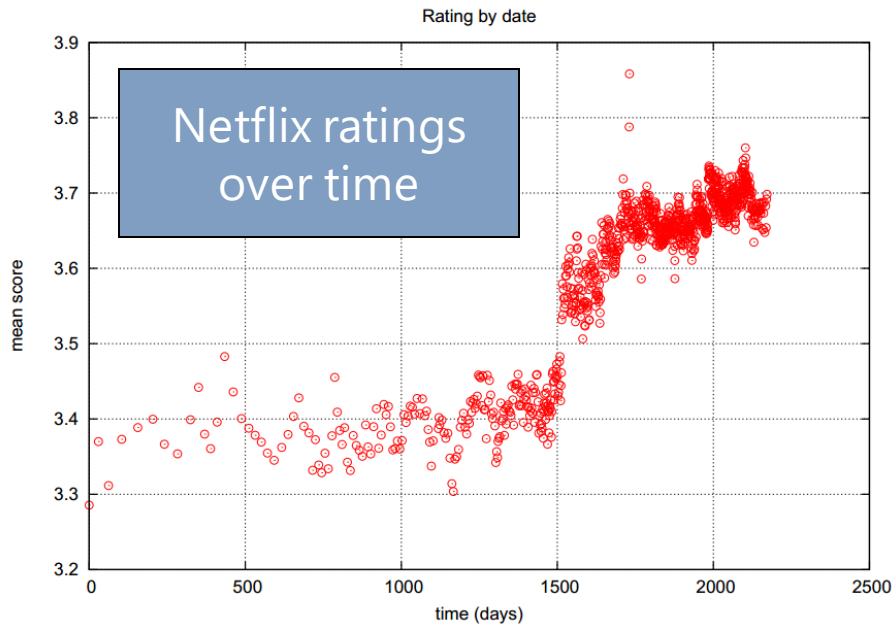
$$\beta_u^{(1)}(t) = \beta_u + \alpha_u \cdot \text{dev}_u(t)$$

overall  
user bias

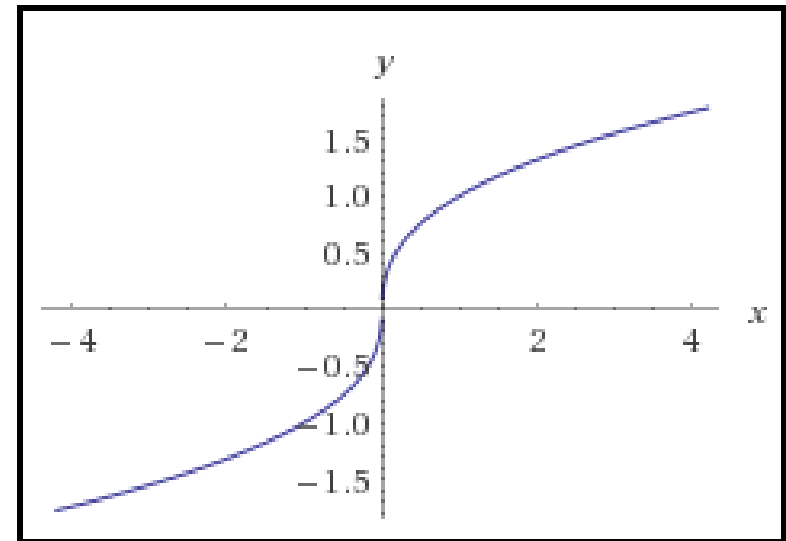
sign and scale for  
deviation term



# Temporal latent-factor models



Real data



Fitted model

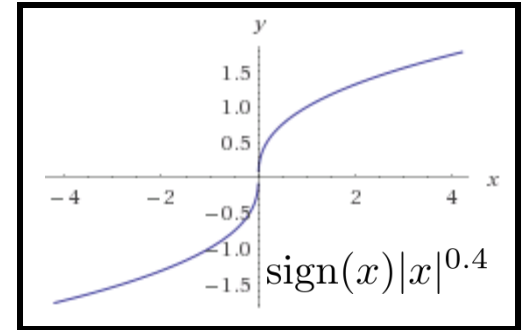
# Temporal latent-factor models

time-dependent user bias can then be defined as:

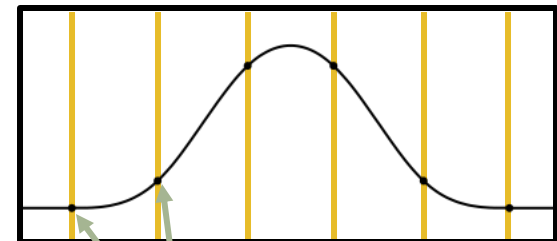
$$\beta_u^{(1)}(t) = \beta_u + \alpha_u \cdot \text{dev}_u(t)$$

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



control points

# Temporal latent-factor models

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

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

user bias associated with this control point

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

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

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:

$$\beta_{u,t}$$

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:

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

global offset  $\alpha$     gradual deviation (or splines)  $\alpha_u \cdot \text{dev}_u(t)$     item bias  $\beta_i$     gradual item bias drift  $\beta_{i,\text{Bin}}(t)$

user bias  $\beta_u$     single-day dynamics  $\beta_{u,t}$

# Temporal latent-factor models

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

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

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

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

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

factor-dependent  
user drift

factor-dependent  
short-term effects



# Temporal latent-factor models

## 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

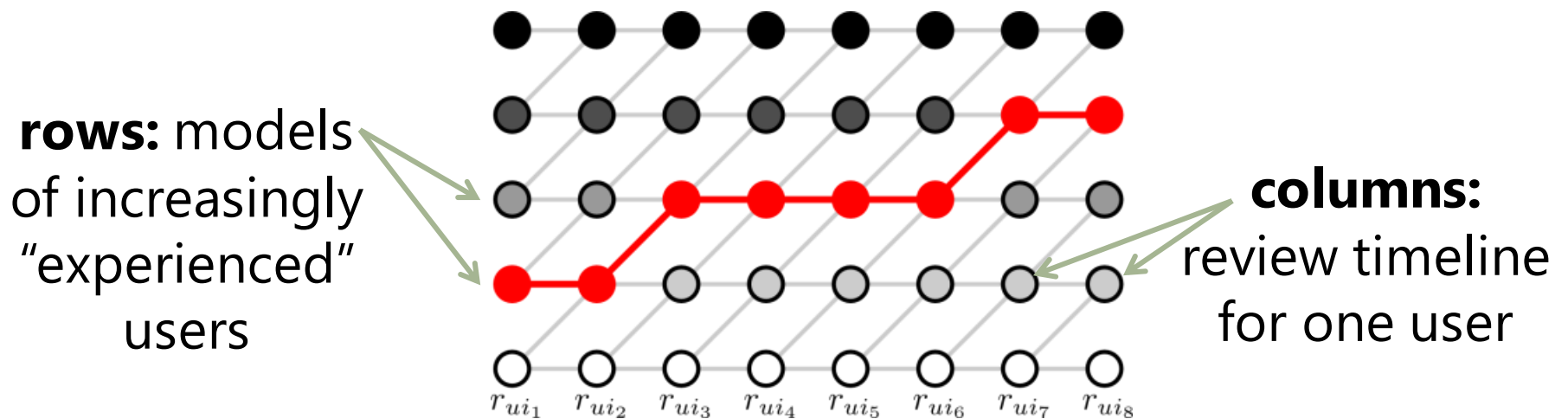
On the other hand...

- 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



# Questions?

Further reading:  
"Collaborative filtering with temporal  
dynamics"

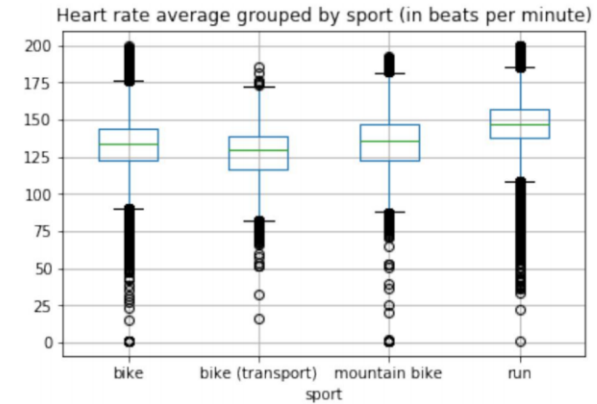
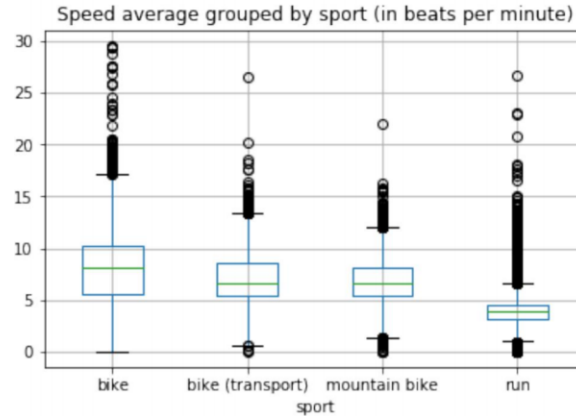
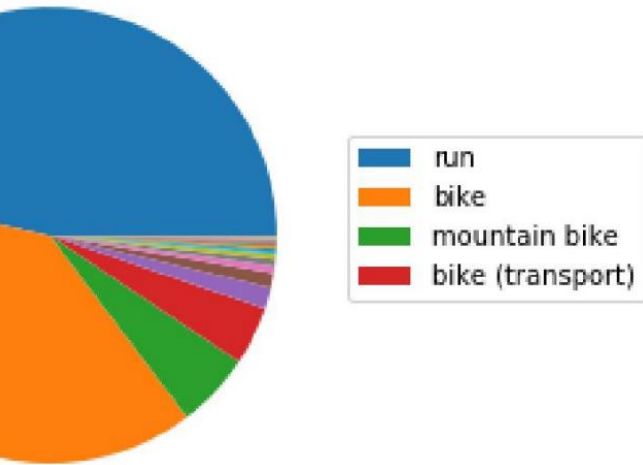
Yehuda Koren, 2009

<http://research.yahoo.com/files/kdd-fp074-koren.pdf>

# Web Mining and Recommender Systems

Incredible assignments

# Predicting Sport Type on EndoMondo



Multiclass classification (four common sport types). Predictive features include:

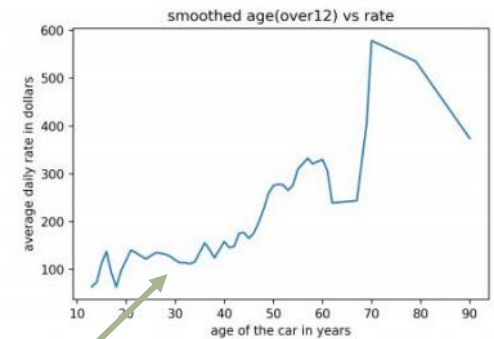
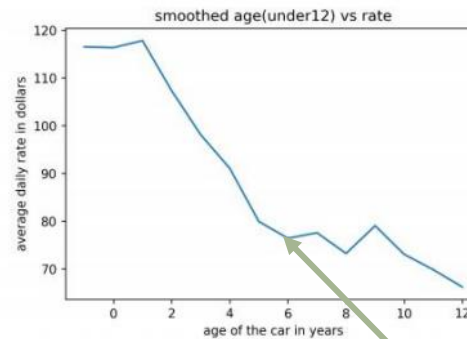
- Altitude (mountain vs. road biking)
- Speed
- Time (e.g. commuting is short)
- *Variation* in speed (e.g. for mountain

Variable	Description
Speed	Recorded speed in Miles per Hour
Altitude	Recorded altitude in Meters
Heart Rate	Recorded heart rate in Beats per Minute
Timestamp	UNIX timestamp
Longitude	Recorded longitude
Latitude	A Recorded latitude
ID	ID of this workout
URL	URL of this workout
User ID	ID of the user
Sport	Type of sport that user engages in
Gender	Male/Female/Unknown

Model	Features	Accuracy	Balanced Accuracy
Logistic Regression	Baseline	0.774	0.435
KNN	Baseline	0.779	0.442
Random Forest	Baseline	0.759	0.454
Logistic Regression	Engineered	0.826	0.465
KNN	Engineered	0.797	0.572
Random Forest	Engineered	<b>0.902</b>	<b>0.705</b>

# Spatially Inspired Price Prediction for Car Rentals

- **Turo** (peer to peer rentals)
- 36,000 rental datapoints from a public github
- Use lat/lon data to extract zipcodes (*uszipcode* library), and combine this with census data from *census.gov* to extract median incomes
- Scrape *Google Trends* listings to determine the popularity of each car



Price vs. car age

Extracted features:

- UserID/carID/rating
- Time to respond to a rental request
- Weekday, month
- Car popularity
- Etc.

Random Forest classifier:

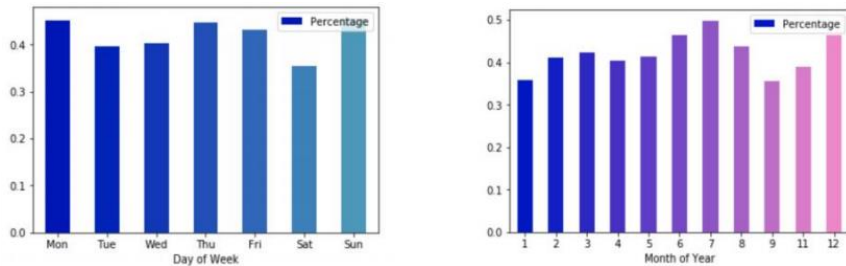
$$R^2 = 0.6115$$

Farhood Ensan  
Kaushik Ganapathy  
Jiaxi Lei

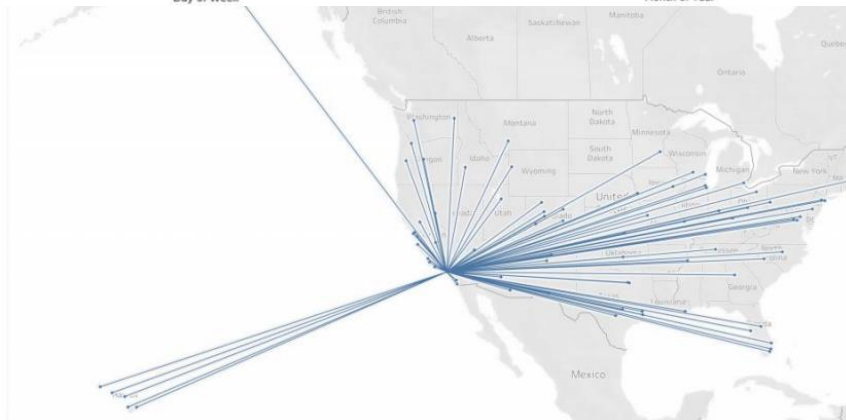
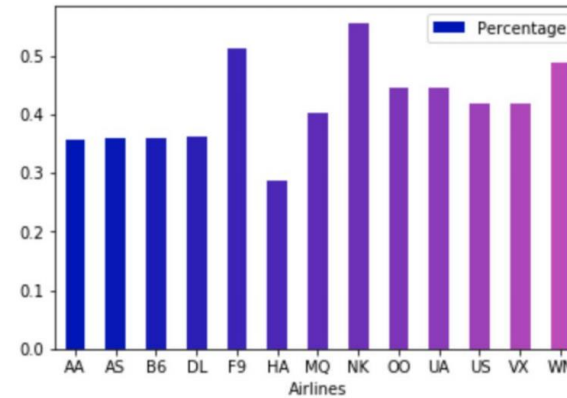
# Airline Flight Delay Prediction

- Predict delays at LAX
- Temporal features, airline features, geographical features
- Accuracy  $\sim 0.65$
- F1  $\sim 0.55$

### Delay vs. day/month



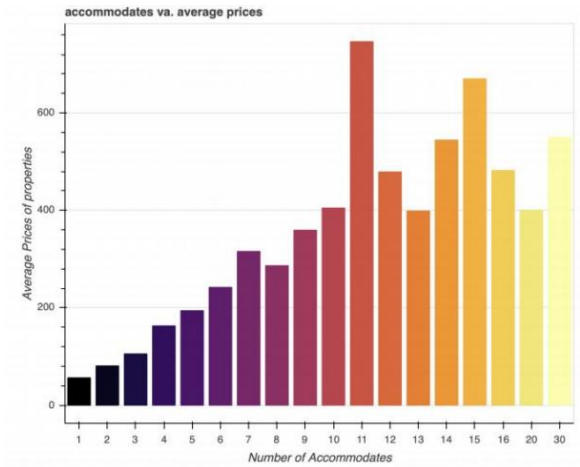
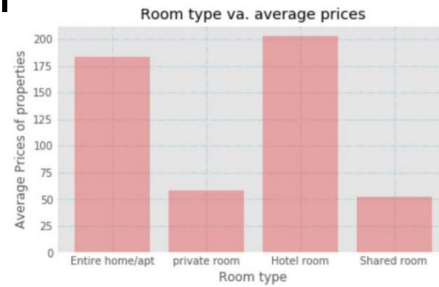
### Delay vs. airline



### Delay vs. destination

# AirBnB Price-Per Prediction

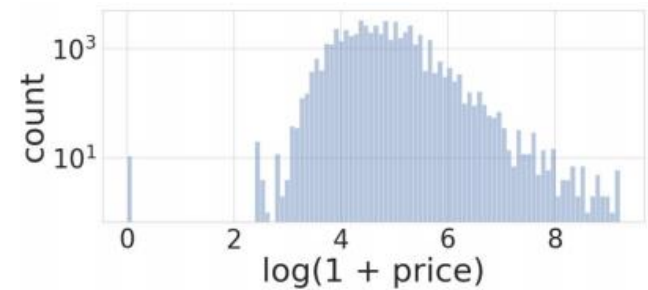
- 45,053 LA AirBnB listings from "Inside AirBnB"
- 85,273 London listings
- 48,895 NY listings



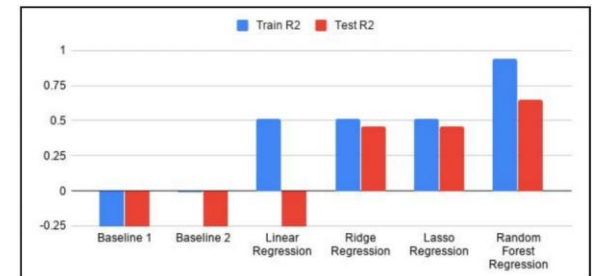
Features include:

- Geo / neighborhood
- Room types / # guests
- Amenities
- Ratings
- Description word-clouds
- Etc.

Number of guests accommodated



Price per neighborhood



Shubham Srivastava

Chang Zhou, Moyan Zhou

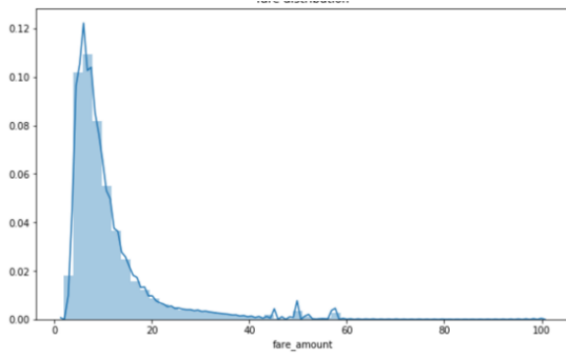
Chi-Chen Lo, Chun-Yi Tu, Sheng-Chuan Chou, Tzu-Wei Sung





# New York City Taxi Fare Prediction

- Predict the total fare of a taxi trip
- 5,000,000 pickup/dropoff datapoints
- MSE and MAPE (Mean Absolute Percentage Error)



Fare distribution



Beidan Huang  
Yixin Zou

Features	Explanation	Usage
AbsLatDiff	Absolute difference in latitude	Baseline, Linear Regression, Random Forest
AbsLonDiff	Absolute difference in longitude	Baseline, Linear Regression, Random Forest
Passenger_count	Number of passengers per ride	Linear Regression, Random Forest
Haversine	Distance metrics taking into account the spherical shape of the Earth	Linear Regression, Random Forest
Fare-bin	Bin range of the fair amount	Upgraded liner regression
Color	Color of the car	
distance	Sphere distance of pickup and drop-off locations	LGBM
bearing	Bearing distance of pickup and drop-off locations	LGBM
Pickup_latitude, pickup_longitude	Pickup location	LGBM
Dropoff_latitude, Dropoff_longitude	Dropoff location	LGBM
Hour, day, month, weekday, year	Hour, day, month, weekday, year of pickup time	LGBM

# Predicting Wave Height using Embedded Sensors on Surfboards

"Smartfin" data from 135 surf sessions

- Accelerometer (A), Gyroscope (G), and Menetometer (M) measurements in x,y,z directions
- "Groundtruth" data collected from CDIP buoy
- 7,000,000 observations!

A1, G1, M1 Means vs. Significant Wave Heights

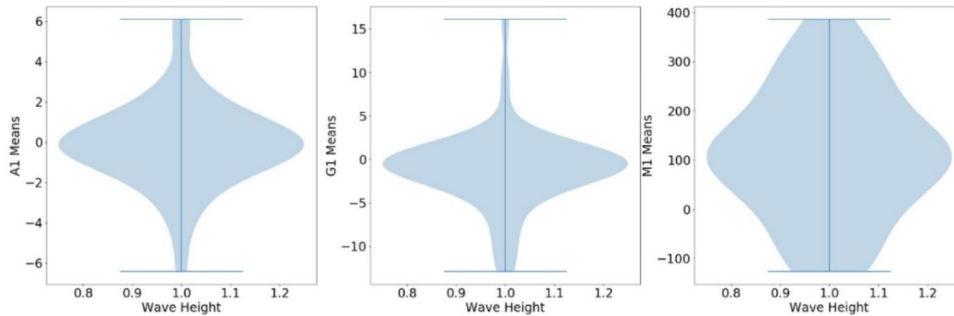
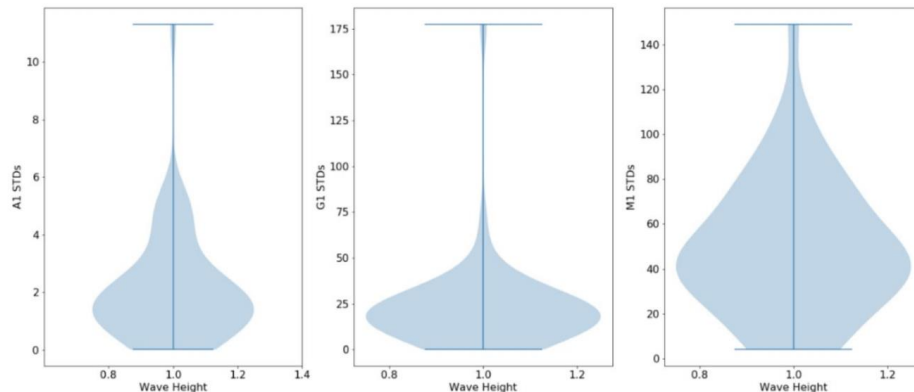
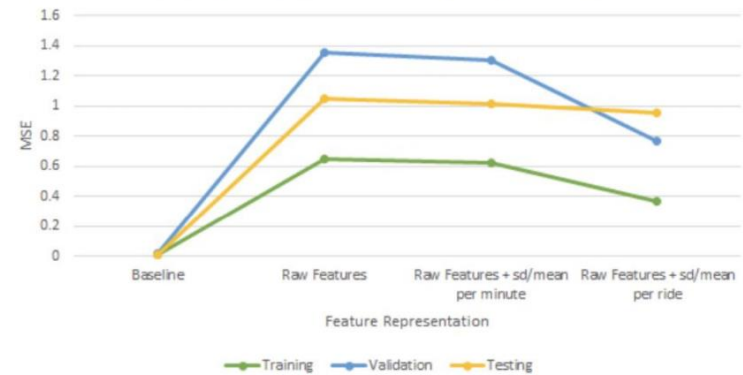


Figure 1: Distribution of A1, G1, and M1 means according to wave height.

A1, G1, M1 STDs vs. Significant Wave Heights



Linear Regression MSE over Feature Representation



Purisa Jasmine Simmons  
Jennifer Chien  
Adrian Salguero  
Martha Gahl

# Course evaluations!

MGT495: <https://academicaffairs.ucsd.edu/Modules/Evals?e5551126>

CSE158: <https://cape.ucsd.edu/students/>

CSE258: <https://academicaffairs.ucsd.edu/Modules/Evals?e5421125>

Thanks!