Online Social Networks and Media

Mining Content

Content

Eduardo J. Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, Alejandro Jaimes: ***Correlating financial time series with micro-blogging activity*.** WSDM 2012: 513-522

Goal

How data from micro-blogging (Twitter) is correlated to time series from the financial domain (prices and traded volume)

Which features from tweets are more correlated with changes in the stocks?

Stock Market Data

Stock data from Yahoo! Finance for 150 (randomly selected) companies in the S&P 500 index for the first half of 2010.

For each stock, the daily closing price and daily traded volume

- Transform the price series into its *daily relative change*, i.e., if the series for price is pi, we used pi – pi-1/pi-1.

- *Normalized traded volume* by dividing the volume of each day by the mean traded volume observed for that company during the entire half of the year.

Twitter Data

Obtain all the *relevant* tweets on the first half of 2010

- Use a series of regular expressions

For example, the filter expression for Yahoo is: “#YHOO | $YHOO | #Yahoo”.

- Manual Refinement

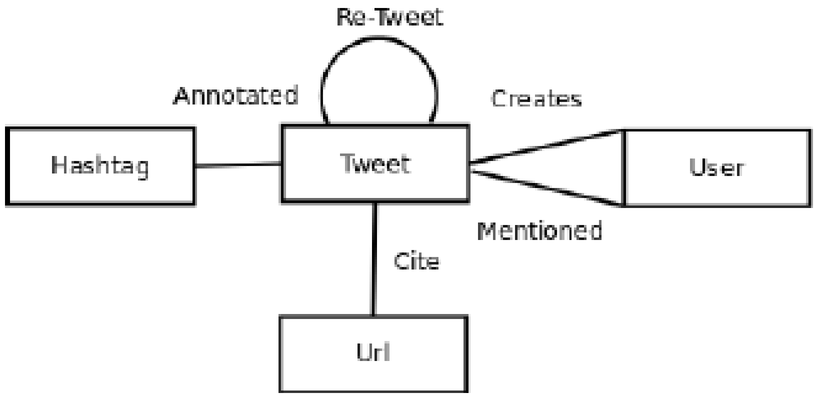
Randomly select 30 tweets from each company, and re-wrote the extraction rules for those sets that had less that 50% of tweets related to the company.

If a rule-based approach not feasible, the company was removed from the dataset

Example companies with expressions rewritten: YHOO, AAPL, APOL

✓ YHOO used in many tweets related with the news service (Yahoo! News).
✓ Apple is a common noun and also used for spamming (“Win a free iPad” scams).
✓ Apollo also the name of a deity in Greek mythology

Graph Representation



Nodes	Schema and description
Tweet	(TweetId, Text, Company, Time) A microblog posting
User	(UserId, Name, #Followers, #Friends, Location, Time) A user that posts a tweet or is mentioned
Url	(Url, ExpandedUrl, Time) A URL included in a tweet
Hashtag	(Hashtag, Time) An annotation used in one tweet

Edges	Schema and description
Annotated	(TweetId, Hashtag, Timestamp) Relate a tweet with one hash-tag
Re-tweeted	(RTId, TweetId, Time) Represents the re-tweet action
Mentioned	(TweetId, UserId, Time) A explicit mention of another user
Cited	(TweetId,Url,Time) Connects a URL with tweets including it
Created	(TweetId, UserId, Time) Connects a tweet with its author

Constrained Subgraph

$G^{c}_{t1,t2}$ about company c at time interval [t1, t2]

induced subgraph of G that contains the nodes that are either *tweets with timestamps in interval [t1,t2]*, or *non-tweet nodes connected* through an edge to the selected tweet nodes.

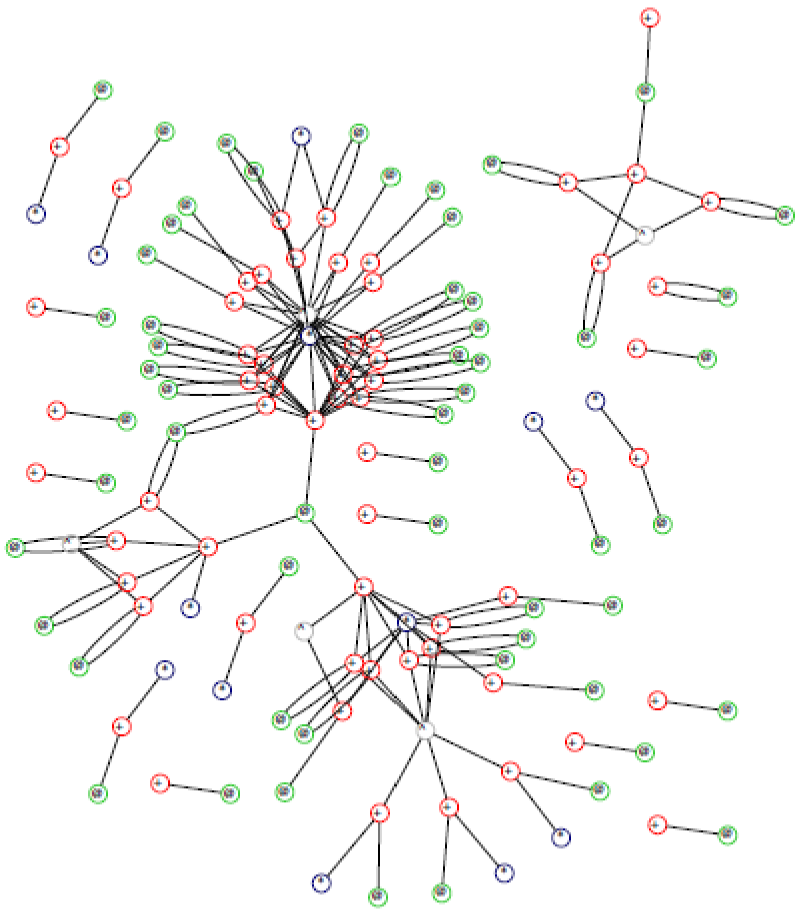


Figure 1: Example of a constrained subgraph for one day and one stock (YHOO). Tweets are presented with red color (+), users are presented with green (@), and URLs with blue (*). Light gray are the similarity nodes (∧)

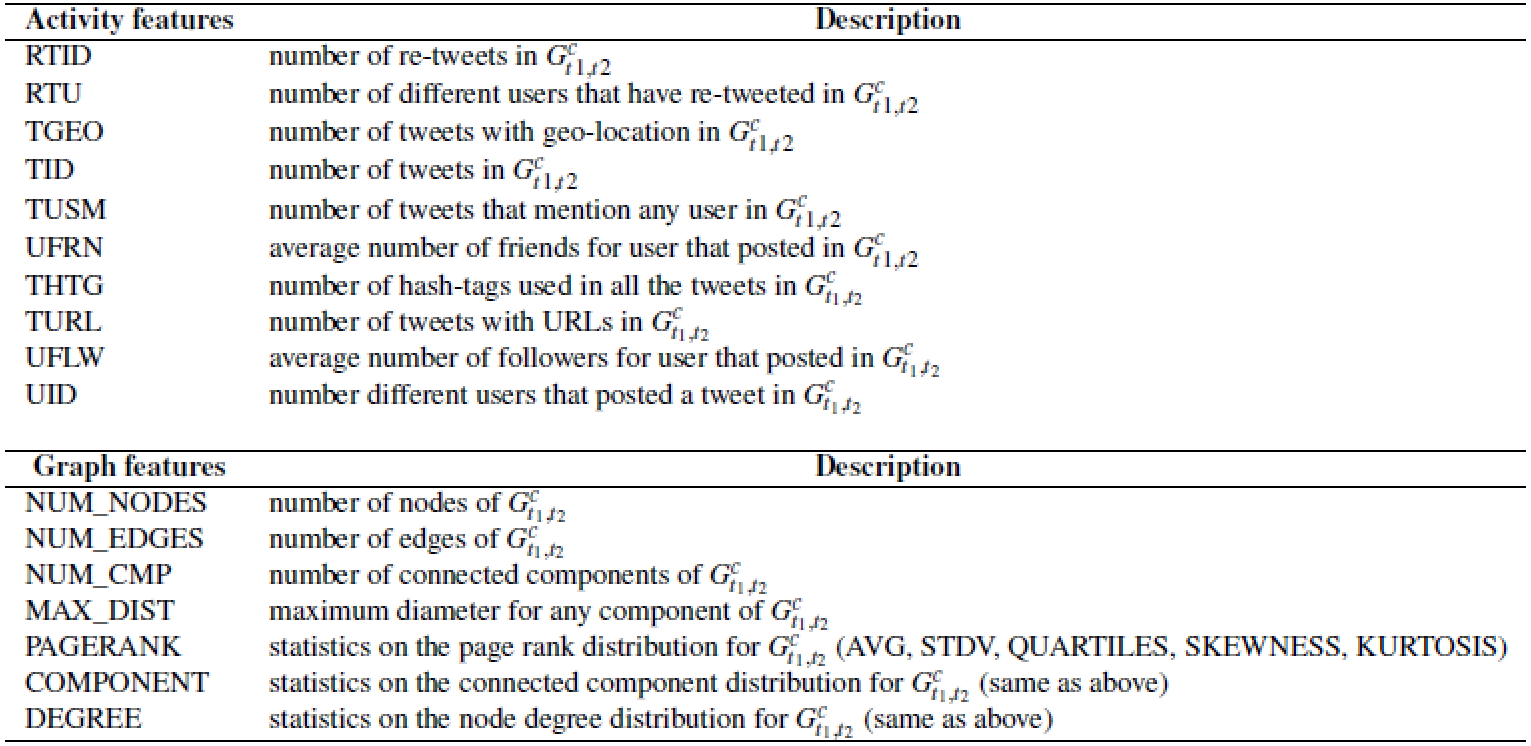
Features

- Activity features count the number of nodes of a particular type, such as number of tweets, number of users, number of hashtags, etc.

- Graph features measure properties of the link structure of the graph.

For scalability, feature computation done using Map-Reduce

Features



Activity features	Description
RTID	number of re-tweets in $G^c_{t1,t2}$
RTU	number of different users that have re-tweeted in $G^c_{t1,t2}$
TGEO	number of tweets with geo-location in $G^c_{t1,t2}$
TID	number of tweets in $G^c_{t1,t2}$
TUSM	number of tweets that mention any user in $G^c_{t1,t2}$
UFRN	average number of friends for user that posted in $G^c_{t1,t2}$
THTG	number of hash-tags used in all the tweets in $G^c_{t_1,t_2}$
TURL	number of tweets with URLs in $G^c_{t_1,t_2}$
UFLW	average number of followers for user that posted in $G^c_{t_1,t_2}$
UID	number different users that posted a tweet in $G^c_{t_1,t_2}$

Graph features	Description
NUM_NODES	number of nodes of $G^c_{t_1,t_2}$
NUM_EDGES	number of edges of $G^c_{t_1,t_2}$
NUM_CMP	number of connected components of $G^c_{t_1,t_2}$
MAX_DIST	maximum diameter for any component of $G^c_{t_1,t_2}$
PAGERANK	statistics on the page rank distribution for $G^c_{t_1,t_2}$ (AVG, STDV, QUARTILES, SKEWNESS, KURTOSIS)
COMPONENT	statistics on the connected component distribution for $G^c_{t_1,t_2}$ (same as above)
DEGREE	statistics on the node degree distribution for $G^c_{t_1,t_2}$ (same as above)

Features normalization and seasonability

Most values normalized in [0, 1]

The number of tweets is increasing and has a *weekly seasonal effect*.

> normalize the feature values *with a time-dependent normalization factor* that considers seasonality, i.e., is proportional to the total number of messages on each day.

Time Series Correlation

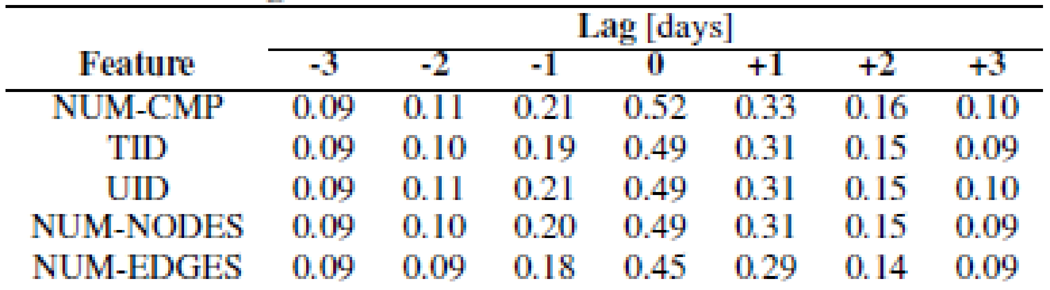
Cross-correlation coefficient (CCF) at lag τ
between series X, Y measures the correlation of the first series with respect to the second series shifted by τ

$$R(\tau) = \frac{\sum_i ((X(i) - \mu_X)(Y(i-\tau) - \mu_Y))}{\sqrt{\sum_i (X(i) - \mu_X)^2}\sqrt{\sum_i (Y(i-\tau) - \mu_Y)^2}}$$

If correlation at a negative lag, then input features can be used to predict the outcome series

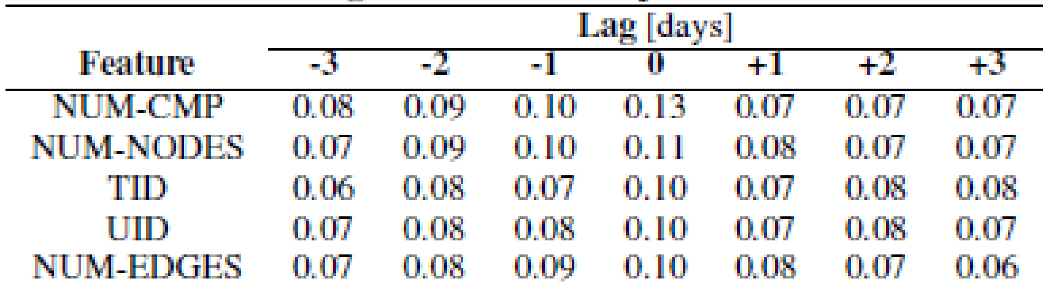
Results

Table 4: Average correlation of traded volume and features.



Feature	Lag [days]						
	-3	-2	-1	0	+1	+2	+3
NUM-CMP	0.09	0.11	0.21	0.52	0.33	0.16	0.10
TID	0.09	0.10	0.19	0.49	0.31	0.15	0.09
UID	0.09	0.11	0.21	0.49	0.31	0.15	0.10
NUM-NODES	0.09	0.10	0.20	0.49	0.31	0.15	0.09
NUM-EDGES	0.09	0.09	0.18	0.45	0.29	0.14	0.09

Table 5: Average correlation of price and features.



Feature	Lag [days]						
	-3	-2	-1	0	+1	+2	+3
NUM-CMP	0.08	0.09	0.10	0.13	0.07	0.07	0.07
NUM-NODES	0.07	0.09	0.10	0.11	0.08	0.07	0.07
TID	0.06	0.08	0.07	0.10	0.07	0.08	0.08
UID	0.07	0.08	0.08	0.10	0.07	0.08	0.07
NUM-EDGES	0.07	0.08	0.09	0.10	0.08	0.07	0.06

Results

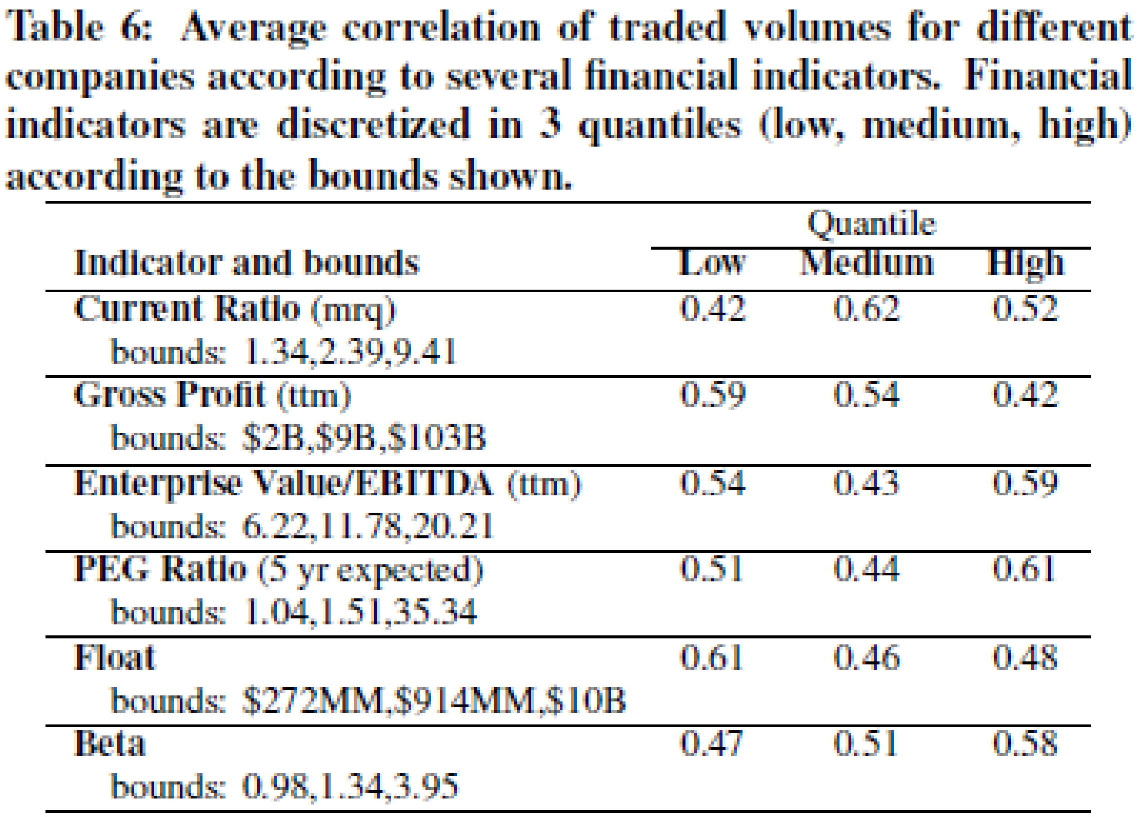


Table 6: Average correlation of traded volumes for different companies according to several financial indicators. Financial indicators are discretized in 3 quantiles (low, medium, high) according to the bounds shown.

	Quantile		
Indicator and bounds	**Low**	**Medium**	**High**
Current Ratio (mrq) bounds: 1.34,2.39,9.41	0.42	0.62	0.52
Gross Profit (ttm) bounds: \$2B,\$9B,\$103B	0.59	0.54	0.42
Enterprise Value/EBITDA (ttm) bounds: 6.22,11.78,20.21	0.54	0.43	0.59
PEG Ratio (5 yr expected) bounds: 1.04,1.51,35.34	0.51	0.44	0.61
Float bounds: \$272MM,\$914MM,\$10B	0.61	0.46	0.48
Beta bounds: 0.98,1.34,3.95	0.47	0.51	0.58

Results

Index graph with data related to the 20 biggest companies (appropriately weighted)

Centrality measures (PageRank, Degree) work better

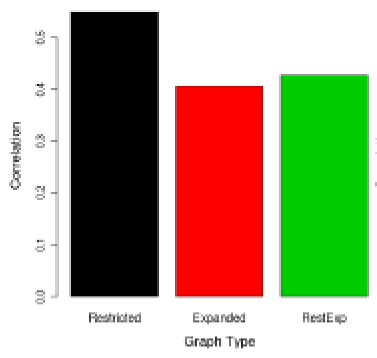
Expanding the graph

Restricted Graph

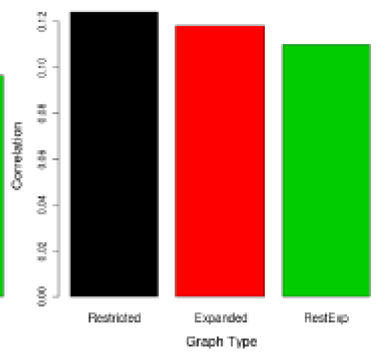
Expanded Graph: all tweets that contain $ticker or #ticker, the full name of the company, short name version after removing common suffixes (e.g., inc or corp), or short name as a hash-tag. Example: “#YHOO | $YHOO | #Yahoo | Yahoo | Yahoo Inc”.

RestExp: Add to the restricted graph the tweets of the expanded graph that are reachable from the nodes of the restricted graph through a path (e.g., through a common author or a re-tweet).

NUM_COMP



(a) Traded volume.



(b) Price change.

Simulation

Goal: simulate daily trading to see if using twitter helps

Description of the Simulator

An investor

1. starts with an initial capital endowment C_0.
2. in the morning of every day t, buys K different stocks using all of the available capital C_t using a number of stock selection strategies
3. holds the stocks all day
4. sells all the stocks at the closing time of day t. The amount obtained is the new capital C_{t+1} used again in step 2.

This process finishes on the last day of the simulation.

Plot the percent of money win or lost each day against the original investment.

Stock selection strategies

Random: buys K stocks at *random*, spends Ct/K per stock (uniformly shared).

Fixed: buys K stocks using a *particular financial indicator* (market capitalization, company size, total debt), from the same companies every day, spends Ct/K per stock(uniformly shared).

Auto Regression: buys the K stocks whose *price changes* will be larger, predicted using an auto-regression (AR(s)) model.

$$x_t = a_1 x_{t-1} + a_2 x_{t-1} \ldots a_m x_{t-m} + c$$

spends Ct/K per stock(uniformly shared) or use a price-weight ratio

$$weight = \frac{\text{price difference}}{\text{open price}}$$

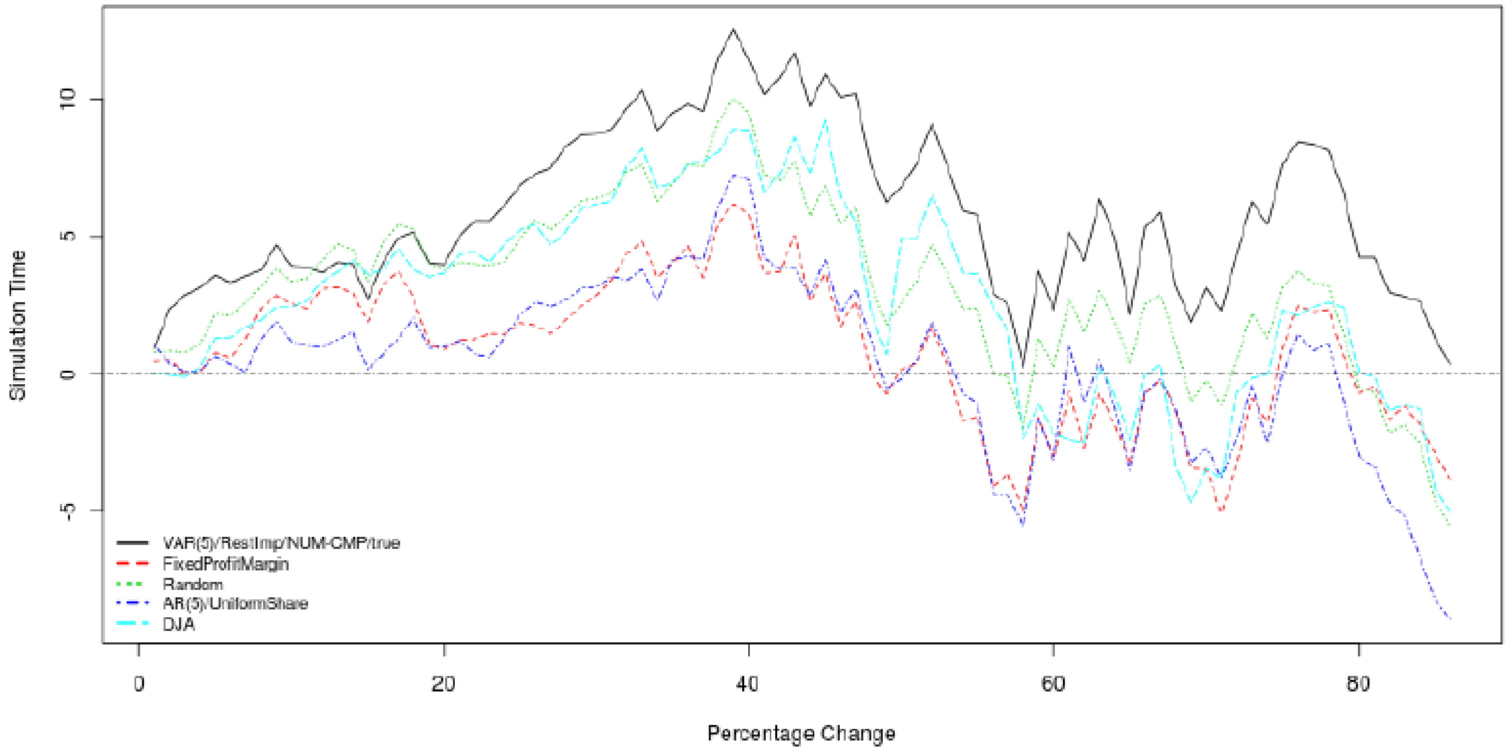
Stock selection strategies

Twitter-Augmented Regression: buys the best K stocks that are predicted using a *vector auto-regressive (VAR(s))* model that considers, in addition to price, a Twitter feature

$$\begin{aligned} x_t = a_1 x_{t-1} + a_2 x_{t-2} \ldots a_m x_{t-m} + \\ b_1 y_{t-1} + b_2 y_{t-2} \ldots b_m y_{t-m} + c. \end{aligned}$$

spends Ct/K per stock(uniformly shared) or use a price-weight ratio

Results



average loss for **Random** is -5.52%, for **AR** -8.9% (Uniform) and -13.08% (Weighted), for **Profit Margin** - 3.8%, Best use **NUN-CMP on RestExp** with uniform share + 0.32% (on restricted graph -2.4% loss)

Includes tDow Jones Index he Average (DJA) (consistent)

Summary

- Present filtering methods to create graphs of postings about a company during a time interval and a suite of features that can be computed from these graphs

- Study the *correlation of the proposed features with the time series of stock price and traded volume* also show how these correlations can be stronger or weaker depending on financial indicators of companies (e.g., on current level of debt)

- Perform a study on the application of the correlation patterns found to guide a stock trading strategy and show that it can lead to a strategy that is competitive when compared to other automatic trading strategies

Takeshi Sakaki, Makoto Okazaki, Yutaka Matsuo: ***Earthquake shakes Twitter users: real-time event detection by social sensors***. WWW 2010: 851-860

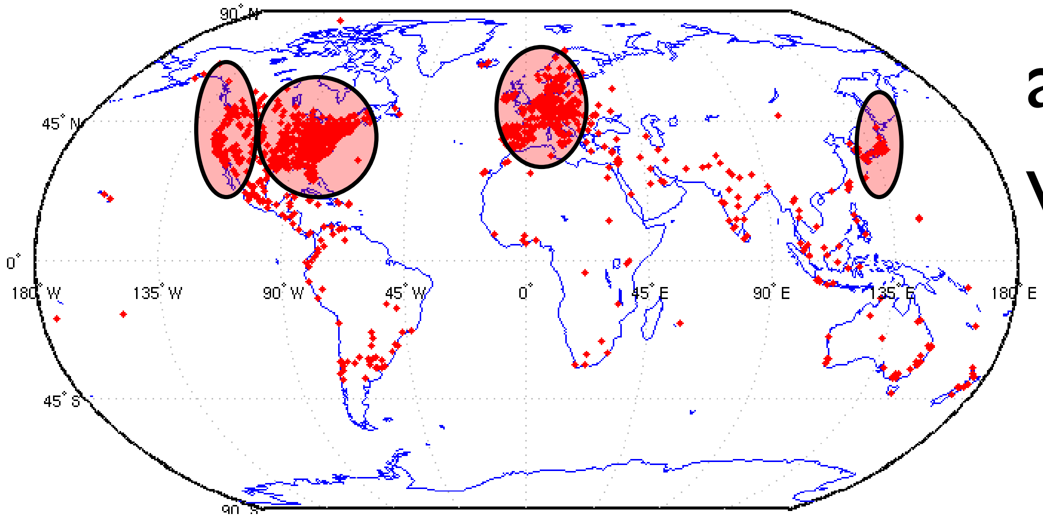
Slides based on the authors' presentation

Goal

- investigate the real-time interaction of events such as earthquakes, in Twitter, and

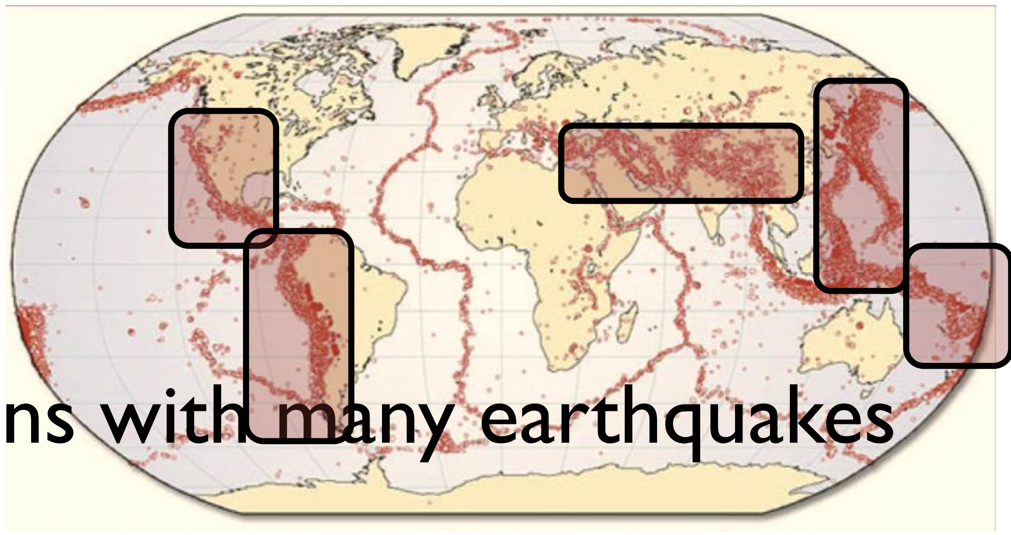
- propose an algorithm to monitor tweets and to detect a target event.

Twitter and Earthquakes in Japan



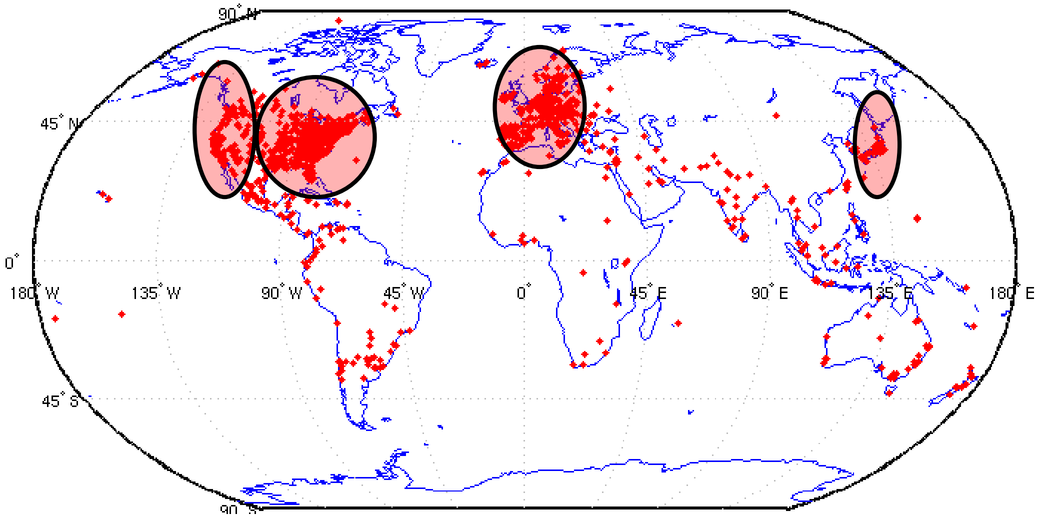
a map of Twitter user world wide

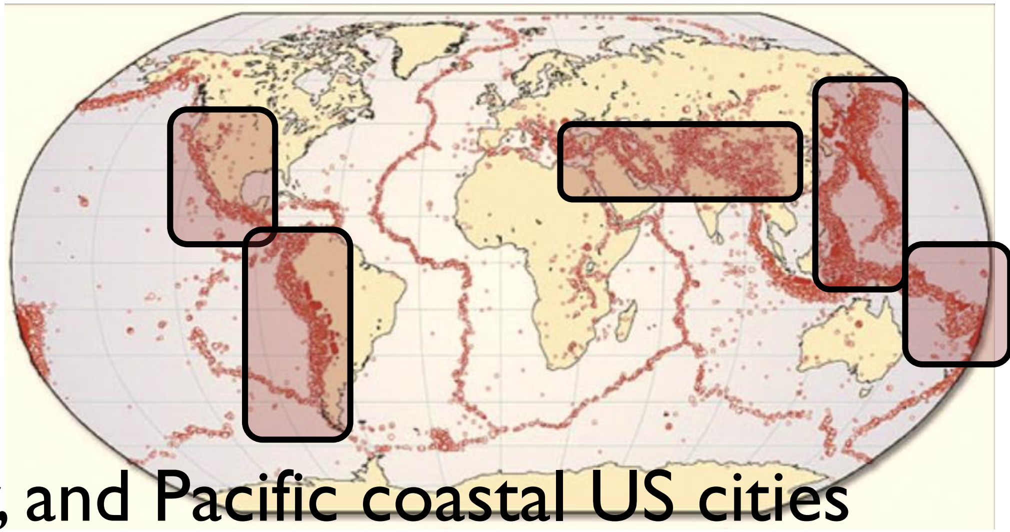
a map of earthquake occurrences world wide



The intersection is regions with many earthquakes and large twitter users.

Twitter and Earthquakes in Japan





Other regions:

Indonesia, Turkey, Iran, Italy, and Pacific coastal US cities

Events

What is an event?

an arbitrary classification of a space/time region.

Example social events: large parties, sports events, exhibitions, accidents, political campaigns.
Example natural events: storms, heavy rainfall, tornadoes, typhoons/hurricanes/cyclones, earthquakes.

Several properties:

I. large scale (many users experience the event),
II. influence daily life (for that reason, many tweets)
III. have spatial and temporal regions (so that real-time location estimation would be possible).

Event detection algorithms

- do *semantic analysis* on tweets
  - to obtain tweets on the target event precisely
- regard Twitter user as a *sensor*
  - to *detect the target event*
  - to *estimate location* of the target

Semantic Analysis on Tweets

- Search tweets including keywords related to a target event – *query keywords*
  - Example: In the case of earthquakes
    - “shaking”, “earthquake”
- Classify tweets into a positive class *(real time reports of the event)* or a negative class
  - Example:
    - “Earthquake right now!!” ---positive
    - “Someone is shaking hands with my boss” --- negative
    - “Three earthquakes in four days. Japan scares me” --- negative
- Build a *classifier*

Semantic Analysis on Tweets

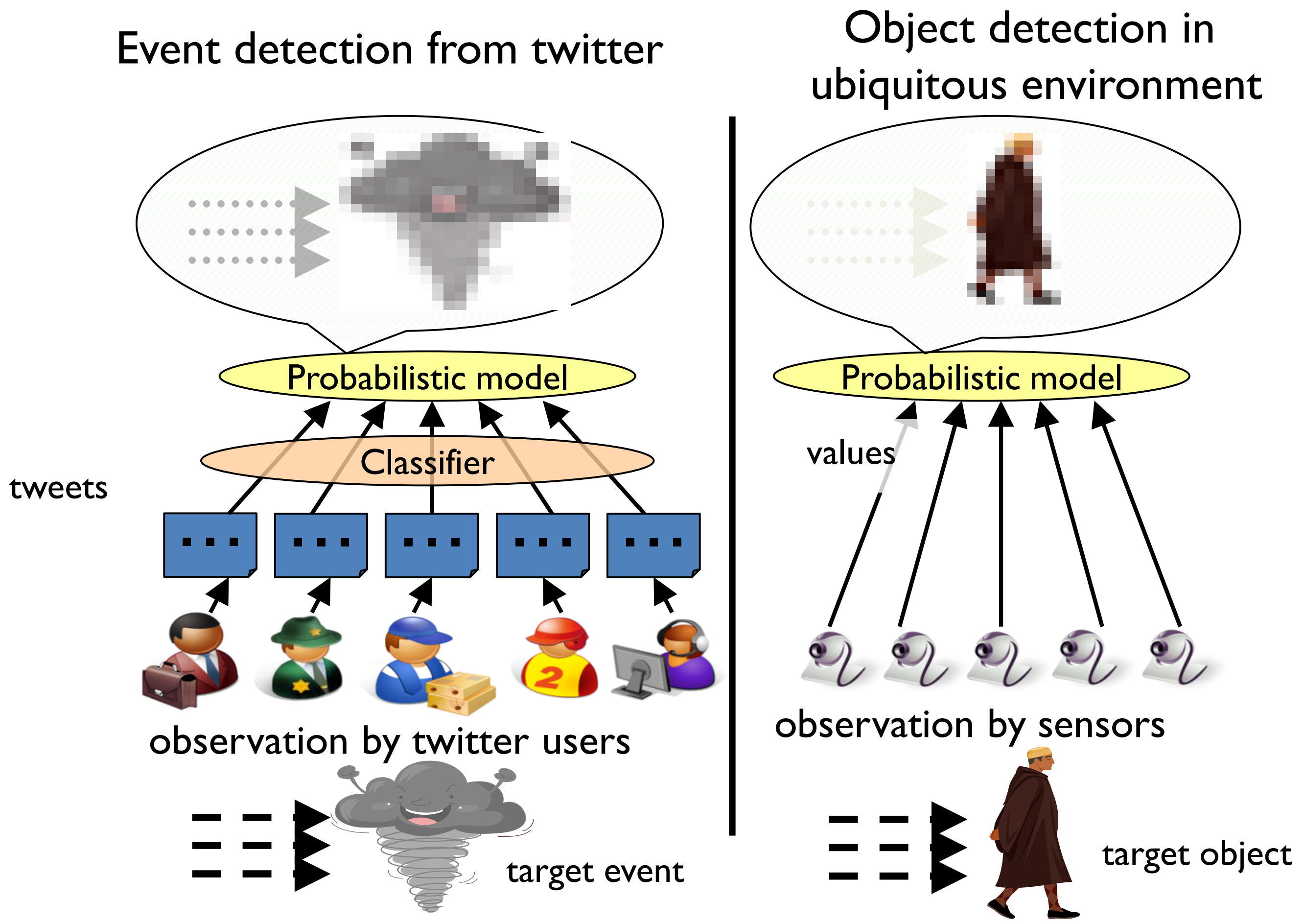
- Create classifier for tweets
  - use Support Vector Machine (SVM)
- Features (Example: I am in Japan, earthquake right now!)
  - Statistical features (A) (7 words, the 5th word)

    the number of words in a tweet message and the position of the query within a tweet
  - Keyword features (B) (I, am, in, Japan, earthquake, right, now)

    the words in a tweet
  - Word context features (C) (Japan, right)

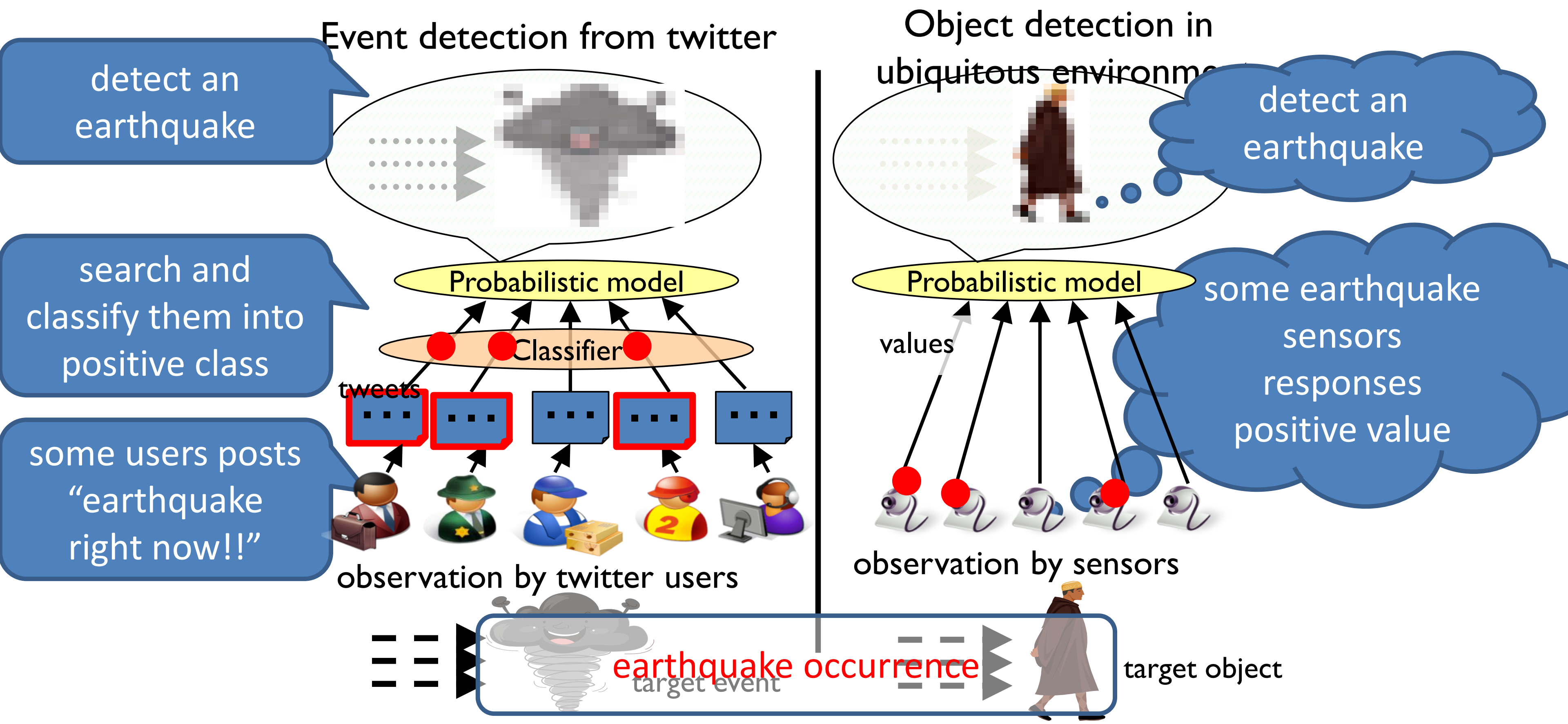
    the words before and after the query word

Tweet as a Sensory Value



the correspondence between *tweets processing* and *sensory data detection*

Tweet as a Sensory Value


detect an earthquake
search and classify them into positive class
some users posts "earthquake right now!!"
Event detection from twitter
Probabilistic model
Classifier
tweets
observation by twitter users
Object detection in ubiquitous environment
detect an earthquake
Probabilistic model
values
some earthquake sensors responses positive value
observation by sensors
earthquake occurrence
target event
target object

We can apply methods for sensory data detection to tweets processing

Tweet as a Sensory Value

- We make two assumptions to apply methods for observation by sensors

- Assumption 1: Each Twitter user is regarded as a sensor
  - a tweet → a sensor reading
  - a sensor detects a target event and makes a report probabilistically
  - Example:
    - make a tweet about an earthquake occurrence
    - “earthquake sensor” return a positive value

- Assumption 2: Each tweet is associated with a time and location
  - a time : post time
  - location : GPS data or location information in user’s profile

Processing time information and location information, we can detect target events and estimate location of target events

Probabilistic Model

- *Why a probabilistic model?*
  - Sensor values (tweets) are noisy and sometimes sensors work incorrectly
  - We cannot judge whether a target event occurred or not from one tweet
  - We have to calculate the probability of an event occurrence from a series of data

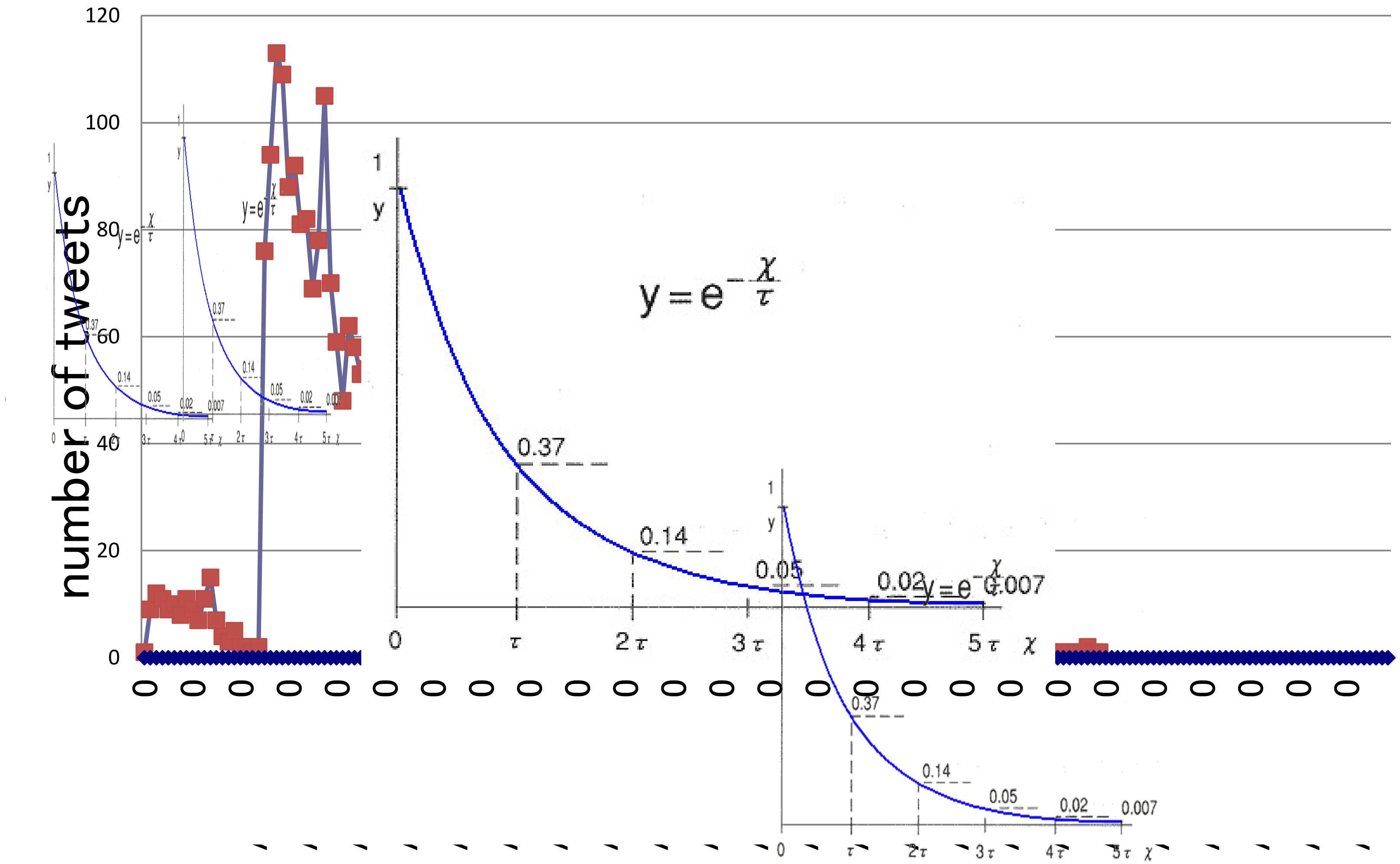
- We propose probabilistic models for
  - detecting events from time-series data
  - estimating the location of an event from sensor readings

Temporal Model

- We must calculate the probability of an event occurrence from multiple sensor values

- We examine the actual time-series data to create a temporal model

Temporal Model



Temporal Model

- the data fits very well to an exponential function with probability density function

$$f(t;\lambda) = \lambda e^{-\lambda t} \left(t > 0, \lambda > 0\right) \qquad \lambda = 0.34$$

- Inter-arrival time (time between events) of a Poisson process, i.e., a process in which events occur continuously and independently at a constant average rate
    - If a user detects an event at time 0, the probability of a tweet from *t to Δt is fixed (λ)*

Temporal Model

- Combine data from many sensors (tweets) based on two assumptions
  - false-positive ratio p_f of a sensor (approximately 0.35)
  - sensors are assumed to be independent and identically distributed (i.i.d.)

The probability of an event occurrence at time t

$$p_{occur}(t) = 1 - p_f^{(n(t))}$$

n(t) total number of sensors (tweets) expected at time t

Temporal Model

- the probability of an event occurrence at time t

$$p_{occur}(t) = 1 - p_f^{n_0\left(1-e^{-\lambda(t+1)}\right)/\left(1-e^{-\lambda}\right)}$$

  - sensors at time 0 → n_0 sensors at time t $n_0 e^{-\lambda t}$
  - the number of sensors at time t $n_0\left(1-e^{-\lambda(t+1)}\right)/\left(1-e^{-\lambda}\right)$

- expected wait time t_{wait} to deliver notification to achieve false positive 1% have to wait for

$$t_{wait} = \left(1-(0.1264/n_0\right)/0.7117 - 1$$

  - parameter $\lambda = 0.34,\ p_f = 0.35,\ p_{occurr} = 0.99$

Location Estimation

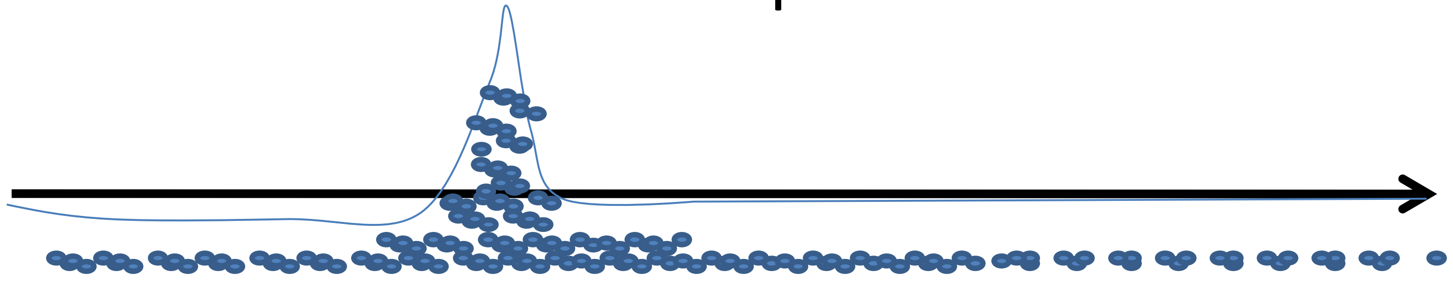
- Compute the target location given a sequence of locations and an i.i.d process noise sequence
- Estimate target recursively

Bayesian Filters for Location Estimation

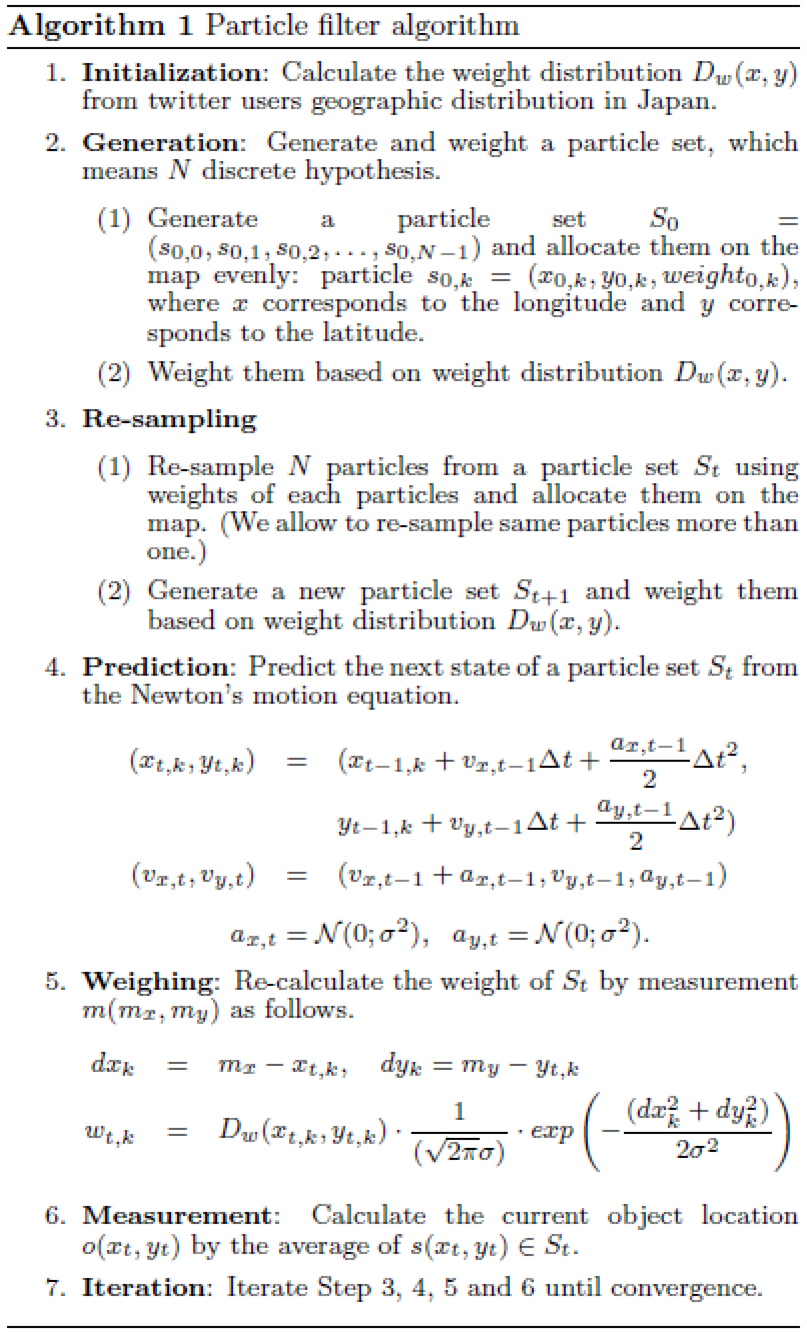
- Kalman Filters
  - are the most widely used variant of Bayes filters
  - Assume that the posterior density at every time is Gaussian, parameterized by a mean and covariance
  - For earthquakes: (longitude, latitude) for typhoons also velocity
  - advantages: computational efficiency
  - disadvantages: being limited to accurate sensors or sensors with high update rates

Particle Filters

- Particle Filters
  - represent the probability distribution by sets of samples, or particles
  - advantages: the ability to represent arbitrary probability densities
    - particle filters can converge to the true posterior even in non-Gaussian, nonlinear dynamic systems.
  - disadvantages: the difficulty in applying to high-dimensional estimation problems



Step 1:
Sample tweets associated with locations and get user distribution proportional to the number of tweets in each region



Algorithm 1 Particle filter algorithm

1. **Initialization**: Calculate the weight distribution $D_w(x, y)$ from twitter users geographic distribution in Japan.
2. **Generation**: Generate and weight a particle set, which means N discrete hypothesis.
   - (1) Generate a particle set $S_0 = (s_{0,0}, s_{0,1}, s_{0,2}, \ldots, s_{0,N-1})$ and allocate them on the map evenly: particle $s_{0,k} = (x_{0,k}, y_{0,k}, weight_{0,k})$, where x corresponds to the longitude and y corresponds to the latitude.
   - (2) Weight them based on weight distribution $D_w(x, y)$.
3. **Re-sampling**
   - (1) Re-sample N particles from a particle set S_t using weights of each particles and allocate them on the map. (We allow to re-sample same particles more than one.)
   - (2) Generate a new particle set S_{t+1} and weight them based on weight distribution $D_w(x, y)$.
4. **Prediction**: Predict the next state of a particle set S_t from the Newton's motion equation.

$$\begin{aligned}(x_{t,k}, y_{t,k}) &= (x_{t-1,k} + v_{x,t-1}\Delta t + \frac{a_{x,t-1}}{2}\Delta t^2, \\ &\quad y_{t-1,k} + v_{y,t-1}\Delta t + \frac{a_{y,t-1}}{2}\Delta t^2) \\ (v_{x,t}, v_{y,t}) &= (v_{x,t-1} + a_{x,t-1}, v_{y,t-1}, a_{y,t-1})\end{aligned}$$

$$a_{x,t} = \mathcal{N}(0; \sigma^2), \quad a_{y,t} = \mathcal{N}(0; \sigma^2).$$

5. **Weighing**: Re-calculate the weight of S_t by measurement $m(m_x, m_y)$ as follows.

$$\begin{aligned} dx_k &= m_x - x_{t,k}, \quad dy_k = m_y - y_{t,k} \\ w_{t,k} &= D_w(x_{t,k}, y_{t,k}) \cdot \frac{1}{(\sqrt{2\pi}\sigma)} \cdot exp\left(-\frac{(dx_k^2 + dy_k^2)}{2\sigma^2}\right)\end{aligned}$$

6. **Measurement**: Calculate the current object location $o(x_t, y_t)$ by the average of $s(x_t, y_t) \in S_t$.
7. **Iteration**: Iterate Step 3, 4, 5 and 6 until convergence.

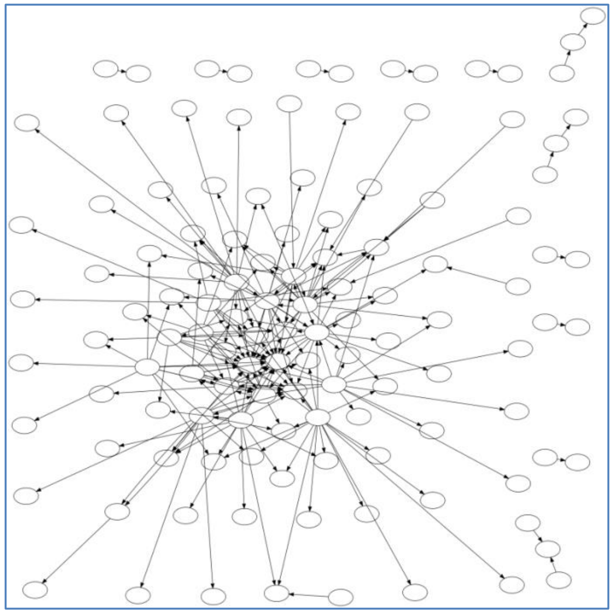
Information Diffusion

- Proposed spatiotemporal models need to meet one condition that
  - Sensors are assumed to be independent

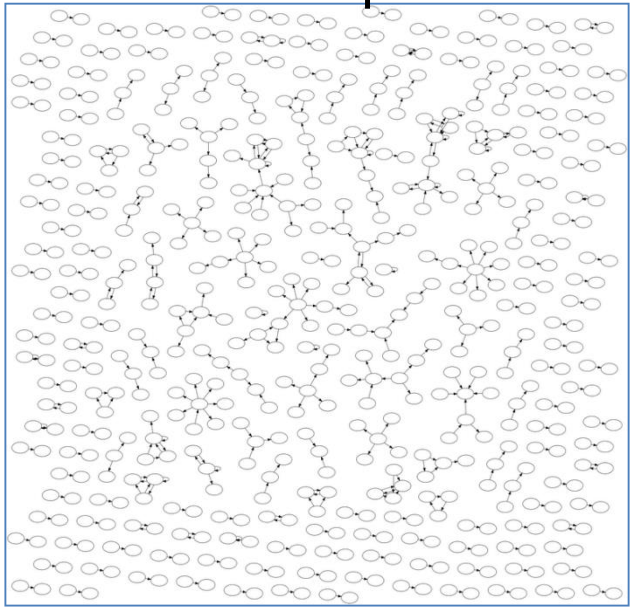
- We check if information diffusion about target events happens because
  - if an information diffusion happened among users, Twitter user sensors are not independent . They affect each other

Information Flow Networks on Twitter

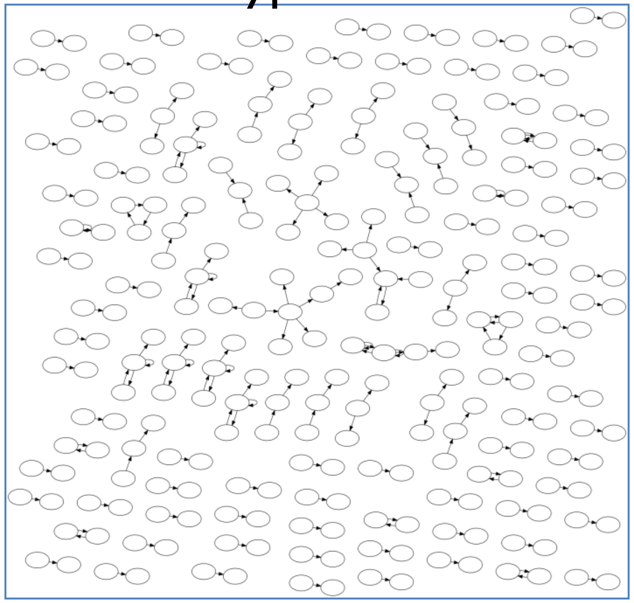
Nintendo DS Game



an earthquake



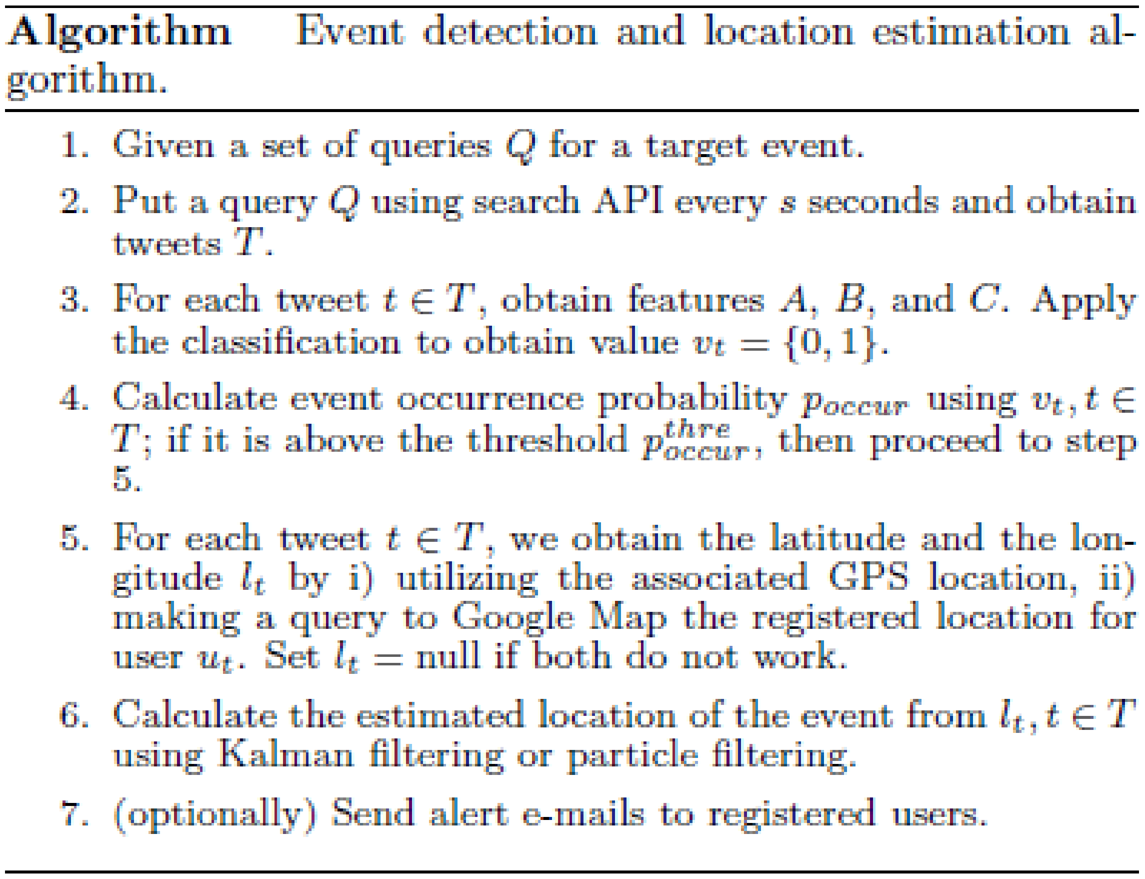
a typhoon



In the case of an earthquakes and a typhoons, *very little information diffusion takes place on Twitter*, compared to Nintendo DS Game

→ We assume that Twitter user sensors are independent about earthquakes and typhoons

General Algorithm



Algorithm Event detection and location estimation algorithm.

1. Given a set of queries Q for a target event.
2. Put a query Q using search API every s seconds and obtain tweets T.
3. For each tweet $t \in T$, obtain features A, B, and C. Apply the classification to obtain value $v_t = \{0, 1\}$.
4. Calculate event occurrence probability p_{occur} using $v_t, t \in T$; if it is above the threshold p_{occur}^{thre}, then proceed to step 5.
5. For each tweet $t \in T$, we obtain the latitude and the longitude l_t by i) utilizing the associated GPS location, ii) making a query to Google Map the registered location for user u_t. Set l_t = null if both do not work.
6. Calculate the estimated location of the event from $l_t, t \in T$ using Kalman filtering or particle filtering.
7. (optionally) Send alert e-mails to registered users.

Evaluation of Semantic Analysis

- Queries
  - Earthquake query: “shaking” and “earthquake”
  - Typhoon query:”typhoon”
- Examples to create classifier
  - 597 positive examples

Evaluation of Semantic Analysis

- "earthquake" query

Features	Recall	Precision	F-Value
Statistical	87.50%	63.64%	73.69%
Keywords	87.50%	38.89%	53.85%
Context	50.00%	66.67%	57.14%
All	87.50%	63.64%	73.69%

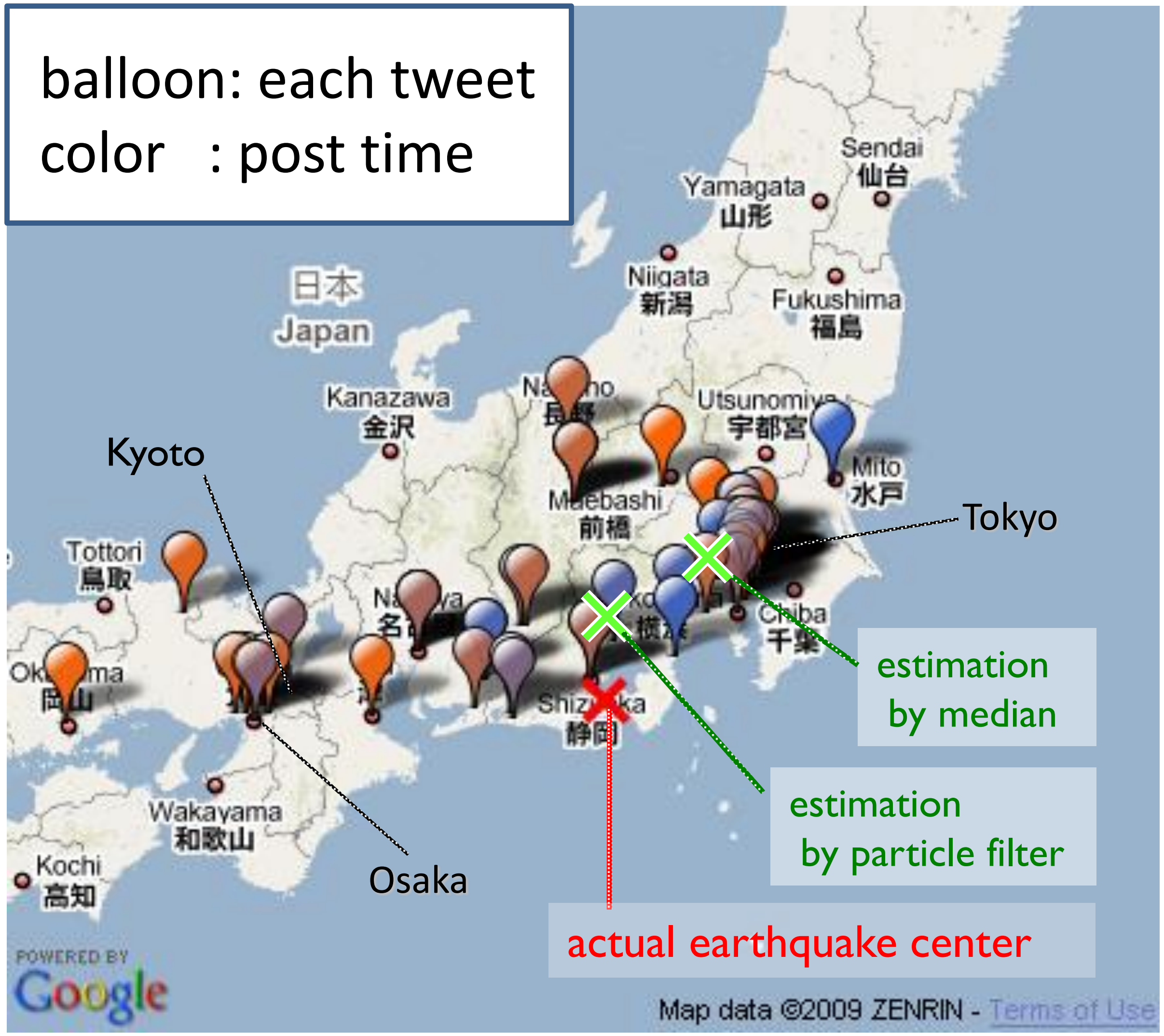
- "shaking" query

Features	Recall	Precision	F-Value
Statistical	66.67%	68.57%	67.61%
Keywords	86.11%	57.41%	68.89%
Context	52.78%	86.36%	68.20%
All	80.56%	65.91%	72.50%

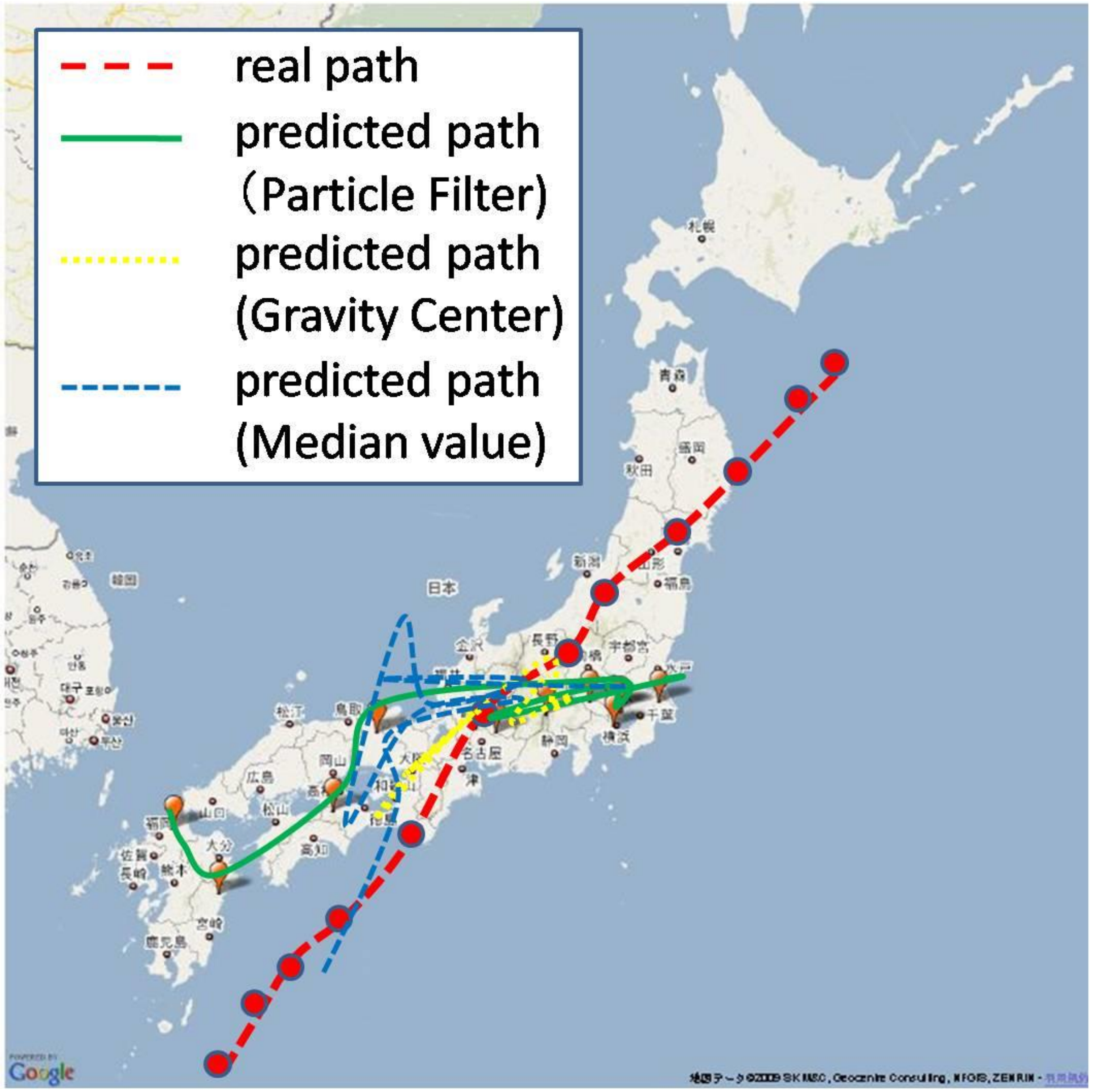
Evaluation of Spatial Estimation

- Target events
  - earthquakes
    - 25 earthquakes from August 2009 to October 2009
  - typhoons
    - name: Melor
- Baseline methods
  - weighed average
    - simply takes the average of latitude and longitude
  - the median
    - simply takes the median of latitude and longitude
- We evaluate methods by distances from actual centers
  - a method works better if the distance from an actual center is smaller

Evaluation of Spatial Estimation

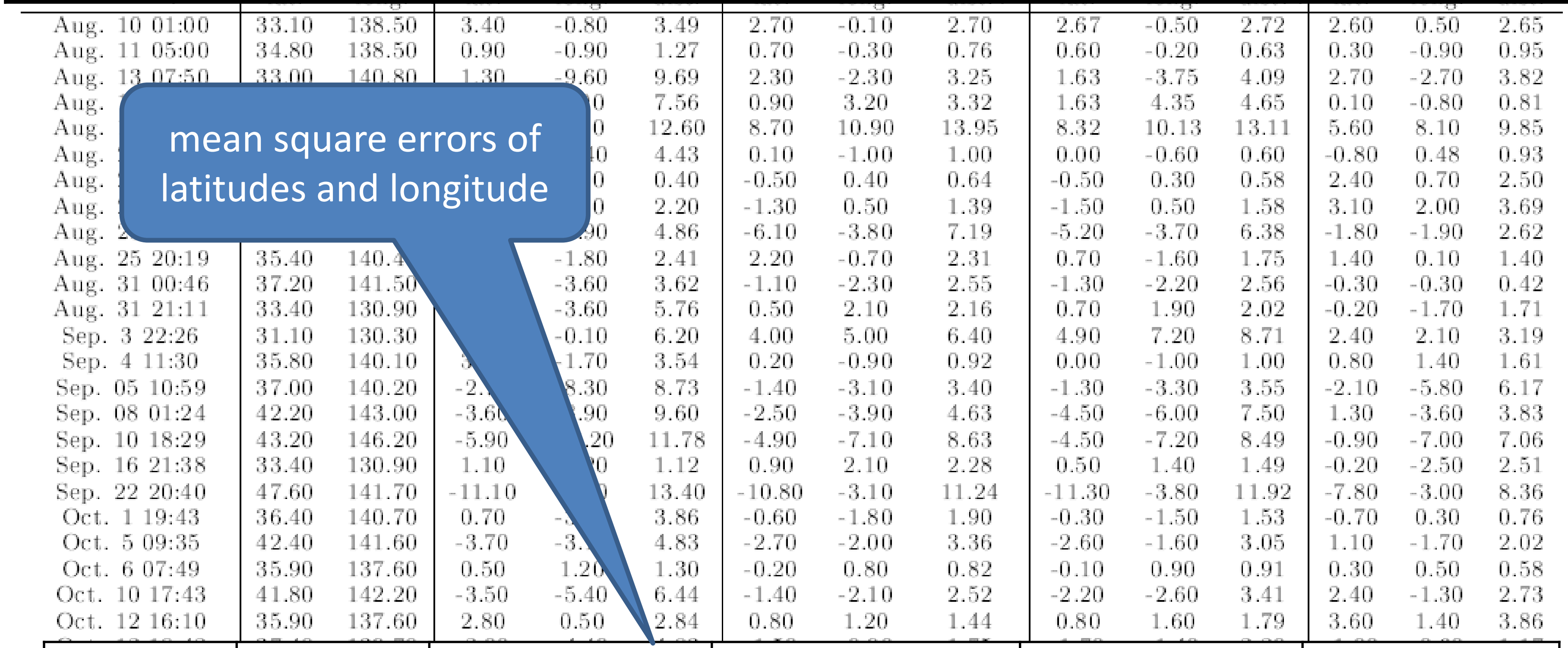


Evaluation of Spatial Estimation


real path
predicted path (Particle Filter)
predicted path (Gravity Center)
predicted path (Median value)

Evaluation of Spatial Estimation

Earthquakes

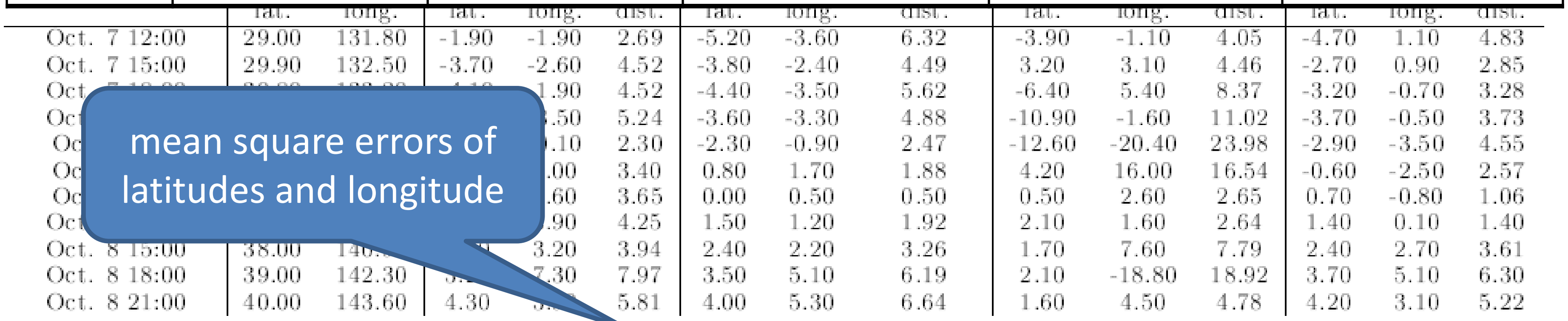


Date	Actual Center		Median			Weighed Average			Kalman Filter			Particle Filter		
Aug. 10 01:00	33.10	138.50	3.40	-0.80	3.49	2.70	-0.10	2.70	2.67	-0.50	2.72	2.60	0.50	2.65
Aug. 11 05:00	34.80	138.50	0.90	-0.90	1.27	0.70	-0.30	0.76	0.60	-0.20	0.63	0.30	-0.90	0.95
Aug. 13 07:50	33.00	140.80	1.30	-9.60	9.69	2.30	-2.30	3.25	1.63	-3.75	4.09	2.70	-2.70	3.82
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7.56	0.90	3.20	3.32	1.63	4.35	4.65	0.10	-0.80	0.81
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	12.60	8.70	10.90	13.95	8.32	10.13	13.11	5.60	8.10	9.85
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4.43	0.10	-1.00	1.00	0.00	-0.60	0.60	-0.80	0.48	0.93
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.40	-0.50	0.40	0.64	-0.50	0.30	0.58	2.40	0.70	2.50
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2.20	-1.30	0.50	1.39	-1.50	0.50	1.58	3.10	2.00	3.69
Aug. [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4.86	-6.10	-3.80	7.19	-5.20	-3.70	6.38	-1.80	-1.90	2.62
Aug. 25 20:19	35.40	[illegible]	[illegible]	-1.80	2.41	2.20	-0.70	2.31	0.70	-1.60	1.75	1.40	0.10	1.40
Aug. 31 00:46	37.20	141.50	[illegible]	-3.60	3.62	-1.10	-2.30	2.55	-1.30	-2.20	2.56	-0.30	-0.30	0.42
Aug. 31 21:11	33.40	130.90	[illegible]	-3.60	5.76	0.50	2.10	2.16	0.70	1.90	2.02	-0.20	-1.70	1.71
Sep. 3 22:26	31.10	130.30	[illegible]	-0.10	6.20	4.00	5.00	6.40	4.90	7.20	8.71	2.40	2.10	3.19
Sep. 4 11:30	35.80	140.10	[illegible]	-1.70	3.54	0.20	-0.90	0.92	0.00	-1.00	1.00	0.80	1.40	1.61
Sep. 05 10:59	37.00	140.20	[illegible]	[illegible]	8.73	-1.40	-3.10	3.40	-1.30	-3.30	3.55	-2.10	-5.80	6.17
Sep. 08 01:24	42.20	143.00	[illegible]	[illegible]	9.60	-2.50	-3.90	4.63	-4.50	-6.00	7.50	1.30	-3.60	3.83
Sep. 10 18:29	43.20	146.20	-5.90	[illegible]	11.78	-4.90	-7.10	8.63	-4.50	-7.20	8.49	-0.90	-7.00	7.06
Sep. 16 21:38	33.40	130.90	1.10	[illegible]	1.12	0.90	2.10	2.28	0.50	1.40	1.49	-0.20	-2.50	2.51
Sep. 22 20:40	47.60	141.70	-11.10	[illegible]	13.40	-10.80	-3.10	11.24	-11.30	-3.80	11.92	-7.80	-3.00	8.36
Oct. 1 19:43	36.40	140.70	0.70	[illegible]	3.86	-0.60	-1.80	1.90	-0.30	-1.50	1.53	-0.70	0.30	0.76
Oct. 5 09:35	42.40	141.60	-3.70	[illegible]	4.83	-2.70	-2.00	3.36	-2.60	-1.60	3.05	1.10	-1.70	2.02
Oct. 6 07:49	35.90	137.60	0.50	1.20	1.30	-0.20	0.80	0.82	-0.10	0.90	0.91	0.30	0.50	0.58
Oct. 10 17:43	41.80	142.20	-3.50	-5.40	6.44	-1.40	-2.10	2.52	-2.20	-2.60	3.41	2.40	-1.30	2.73
Oct. 12 16:10	35.90	137.60	2.80	0.50	2.84	0.80	1.20	1.44	0.80	1.60	1.79	3.60	1.40	3.86
Average	—				5.47			3.62			3.85			3.01

Particle filters works better than other methods

Evaluation of Spatial Estimation

A typhoon



Date	Actual Center		Median			Weighed Average			Kalman Filter			Particle Filter		
	lat.	long.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.
Oct. 7 12:00	29.00	131.80	-1.90	-1.90	2.69	-5.20	-3.60	6.32	-3.90	-1.10	4.05	-4.70	1.10	4.83
Oct. 7 15:00	29.90	132.50	-3.70	-2.60	4.52	-3.80	-2.40	4.49	3.20	3.10	4.46	-2.70	0.90	2.85
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4.52	-4.40	-3.50	5.62	-6.40	5.40	8.37	-3.20	-0.70	3.28
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	5.24	-3.60	-3.30	4.88	-10.90	-1.60	11.02	-3.70	-0.50	3.73
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2.30	-2.30	-0.90	2.47	-12.60	-20.40	23.98	-2.90	-3.50	4.55
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3.40	0.80	1.70	1.88	4.20	16.00	16.54	-0.60	-2.50	2.57
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3.65	0.00	0.50	0.50	0.50	2.60	2.65	0.70	-0.80	1.06
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4.25	1.50	1.20	1.92	2.10	1.60	2.64	1.40	0.10	1.40
Oct. 8 15:00	38.00	[illegible]	[illegible]	3.20	3.94	2.40	2.20	3.26	1.70	7.60	7.79	2.40	2.70	3.61
Oct. 8 18:00	39.00	142.30	[illegible]	7.30	7.97	3.50	5.10	6.19	2.10	-18.80	18.92	3.70	5.10	6.30
Oct. 8 21:00	40.00	143.60	4.30	[illegible]	5.81	4.00	5.30	6.64	1.60	4.50	4.78	4.20	3.10	5.22
Average	**—**				**4.39**			**4.02**			**9.56**			**3.58**

mean square errors of latitudes and longitude

Particle Filters works better than other methods

Discussions of Experiments

- Particle filters performs better than other methods
- If the center of a target event is in the sea, it is more difficult to locate it precisely from tweets
- It becomes more difficult to make good estimation in less populated areas

Earthquake Reporting System

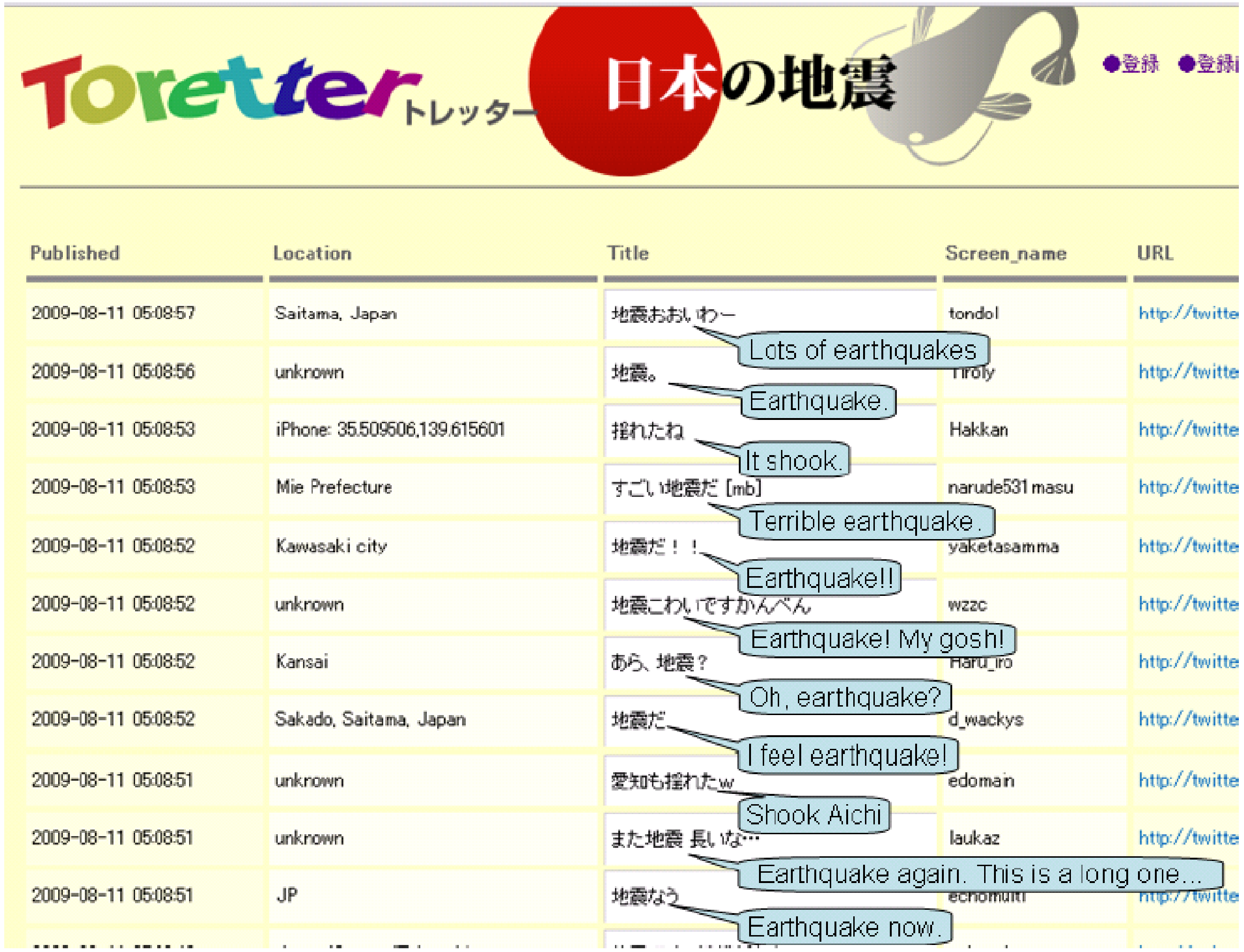
- Toretter (http://toretter.com)
  - Earthquake reporting system using the event detection algorithm
  - All users can see the detection of past earthquakes
  - Registered users can receive e-mails of earthquake detection reports

Dear Alice,

We have just detected an earthquake around Chiba. Please take care.

Toretter Alert System

Screenshot of Toretter.com



Published	Location	Title	Screen_name	URL
2009-08-11 05:08:57	Saitama, Japan	地震おおいわー	tondol	http://twitte
2009-08-11 05:08:56	unkrown	地震。	Tiroly	http://twitte
2009-08-11 05:08:53	iPhone: 35.509506,139.615601	揺れたね	Hakkan	http://twitte
2009-08-11 05:08:53	Mie Prefecture	すごい地震だ [mb]	narude531 masu	http://twitte
2009-08-11 05:08:52	Kawasaki city	地震だ！！	yaketasamma	http://twitte
2009-08-11 05:08:52	unkrown	地震こわいですかんべん	wzzc	http://twitte
2009-08-11 05:08:52	Kansai	あら、地震？	Haru_iro	http://twitte
2009-08-11 05:08:52	Sakado, Saitama, Japan	地震だ	d_wackys	http://twitte
2009-08-11 05:08:51	unkrown	愛知も揺れたw	edomain	http://twitte
2009-08-11 05:08:51	unkrown	また地震 長いな…	laukaz	http://twitte
2009-08-11 05:08:51	JP	地震なう	echomulti	http://twitte

Earthquake Reporting System

- Effectiveness of alerts of this system
  - Alert E-mails urges users to prepare for the earthquake if they are received by a user shortly before the earthquake actually arrives.

- Is it possible to receive the e-mail before the earthquake actually arrives?
  - An earthquake is transmitted through the earth's crust at about 3~7 km/s.
  - a person has about 20~30 sec before its arrival at a point that is 100 km distant from an actual center

Results of Earthquake Detection

Date	Magnitude	Location	Time	E-mail sent time	time gap [sec]	# tweets within 10 minutes	Announce of JMA
Aug. 18	4.5	Tochigi	6:58:55	7:00:30	95	35	7:08
Aug. 18	3.1	Suruga-wan	19:22:48	19:23:14	26	17	19:28
Aug. 21	4.1	Chiba	8:51:16	8:51:35	19	52	8:56
Aug. 25	4.3	Uraga-oki	2:22:49	2:23:21	31	23	2:27
Aug.25	3.5	Fukushima	2:21:15	22:22:29	73	13	22:26
Aug. 27	3.9	Wakayama	17:47:30	17:48:11	41	16	1:7:53
Aug. 27	2.8	Suruga-wan	20:26:23	20:26:45	22	14	20:31
Ag. 31	4.5	Fukushima	00:45:54	00:46:24	30	32	00:51
Sep. 2	3.3	Suruga-wan	13:04:45	13:05:04	19	18	13:10
Sep. 2	3.6	Bungo-suido	17:37:53	17:38:27	34	3	17:43

In all cases, we sent E-mails before announces of JMA
In the earliest cases, we can sent E-mails in 19 sec.

Experiments And Evaluation

- We demonstrate performances of
  - tweet classification
  - event detection from time-series data

    → show this results in “application”
  - location estimation from a series of spatial information

Results of Earthquake Detection

JMA intensity scale	2 or more	3 or more	4 or more
Num of earthquakes	78	25	3
Detected	70 (89.7%)	24 (96.0%)	3 (100.0%)
Promptly detected*	53 (67.9%)	20 (80.0%)	3 (100.0%)

Promptly detected: detected in a minutes
JMA intensity scale: the original scale of earthquakes by Japan Meteorology Agency

Period: Aug.2009 – Sep. 2009
Tweets analyzed : 49,314 tweets
Positive tweets : 6291 tweets by 4218 users

We detected 96% of earthquakes that were stronger than scale 3 or more during the period.

Summary

- We investigated the real-time nature of Twitter for event detection

- Semantic analysis were applied to tweets classification
- We consider each Twitter user as a sensor and set a problem to detect an event based on sensory observations
- Location estimation methods such as Kaman filters and particle filters are used to estimate locations of events

- We developed an earthquake reporting system, which is a novel approach to notify people promptly of an earthquake event
- We plan to expand our system to detect events of various kinds such as rainbows, traffic jam etc.

Jure Leskovec, Lars Backstrom, Jon M. Kleinberg: ***Meme-tracking and the dynamics of the news cycle***. KDD 2009: 497-506

Some slides are from Jure Leskovec's course "On Social Information Network Analysis"

Goal

Track units of information as the evolve over time

How? Extract textual fragments that travel relatively unchanged, through many articles:

> *Look for phrases inside quotes*: **"…"** About 1.25 quotes per document in our data

Why it works?

Quotes are

- integral parts of journalistic practices
- tend to follow iterations of a story as it evolves
- are attributed to individuals and have time and location

Approach

Item: a news article or blog post
Phrase: a quoted string that occurs in one or more items

Produce phrase clusters, which are collections of phrases that are close textual variants of one another.

1. *Build a phrase graph* where each phrase is represented by a node and directed edges connect related phrases.
2. *Partition the graph* in such a way that its components are the phrase clusters.

Phrase Graph

A graph G on the set of quoted phrases:

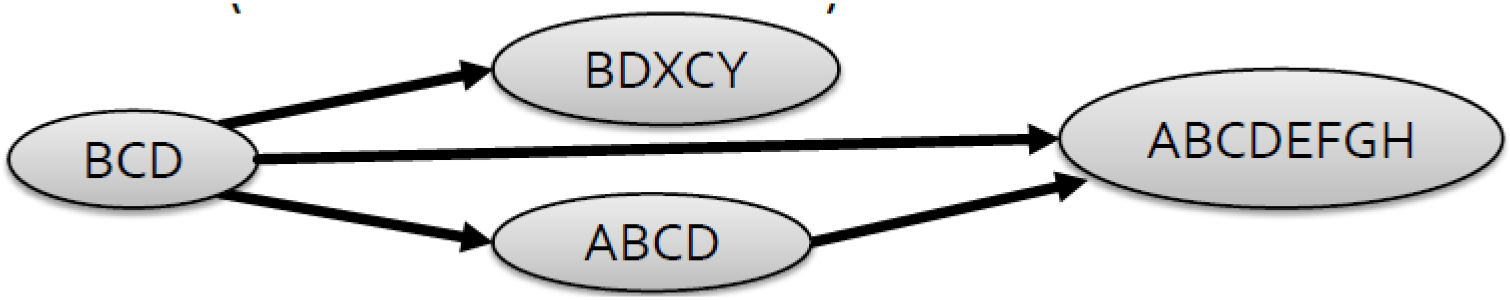
V = phrases

An edge (p, q)

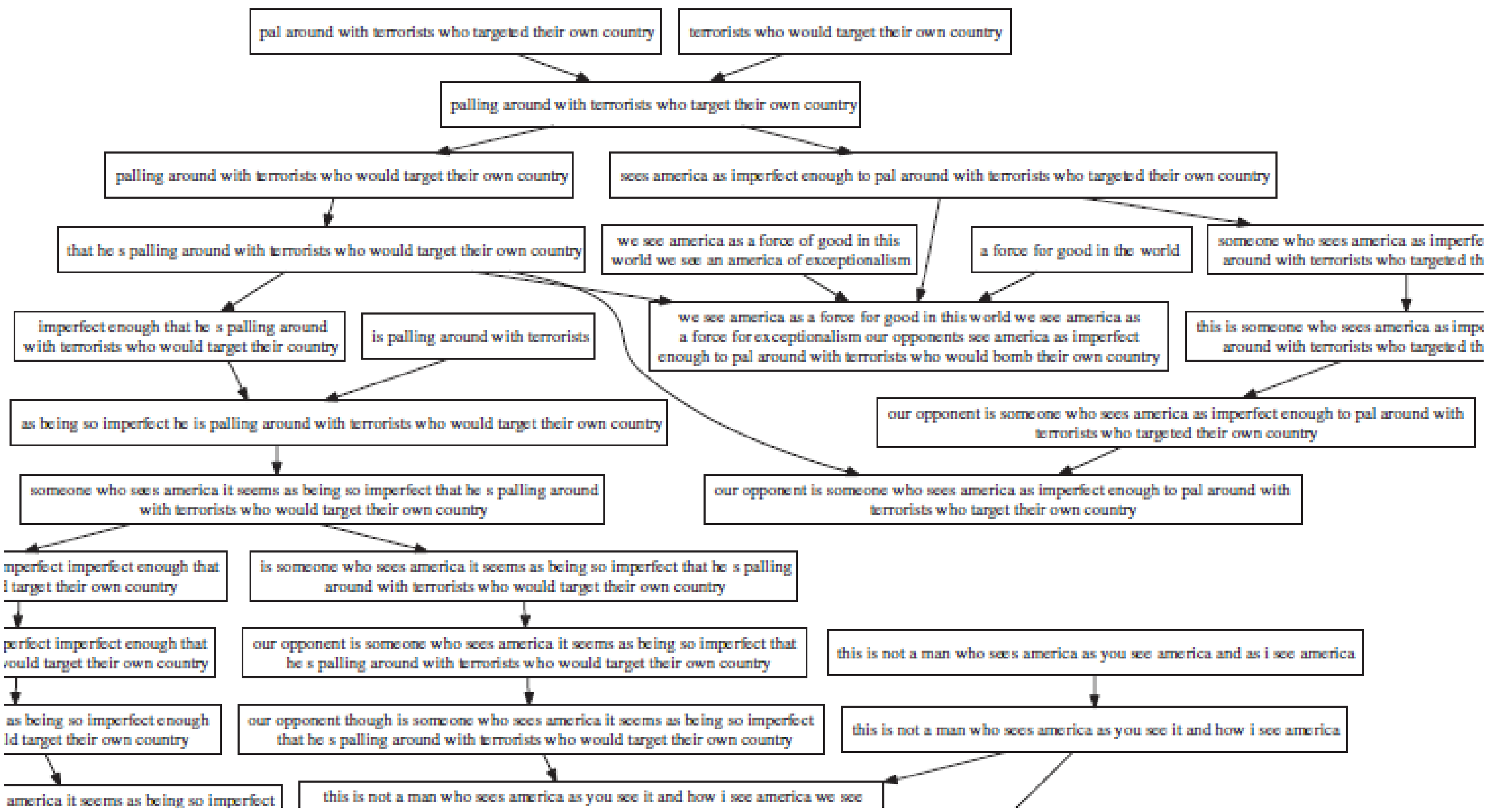
- p is strictly shorter than q, and
- p has directed edit distance to q less than a small threshold or there is at least a k-word consecutive overlap between the phrases

Weights w(p, q): decrease with edit distance from p to q, and increase in the frequency of q in the corpus (the inclusion of p in q is supported by many occurrences of q)

G is a DAG



Phrase Graph


pal around with terrorists who targeted their own country
terrorists who would target their own country
palling around with terrorists who target their own country
palling around with terrorists who would target their own country
sees america as imperfect enough to pal around with terrorists who targeted their own country
that he s palling around with terrorists who would target their own country
we see america as a force of good in this world we see an america of exceptionalism
a force for good in the world
someone who sees america as imperfe around with terrorists who targeted th
imperfect enough that he s palling around with terrorists who would target their country
is palling around with terrorists
we see america as a force for good in this world we see america as a force for exceptionalism our opponents see america as imperfect enough to pal around with terrorists who would bomb their own country
this is someone who sees america as impe around with terrorists who targeted th
as being so imperfect he is palling around with terrorists who would target their own country
our opponent is someone who sees america as imperfect enough to pal around with terrorists who targeted their own country
someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
our opponent is someone who sees america as imperfect enough to pal around with terrorists who target their own country
mperfect imperfect enough that l target their own country
is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
perfect imperfect enough that vould target their own country
our opponent is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
this is not a man who sees america as you see america and as i see america
as being so imperfect enough ld target their own country
our opponent though is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
this is not a man who sees america as you see it and how i see america
america it seems as being so imperfect
this is not a man who sees america as you see it and how i see america we see

Quote: Our opponent is someone who sees America, it seems, as being so imperfect, imperfect enough that he's palling around with terrorists who would target their own country."

Phrase Graph Construction

Preprocessing:

- a lower bound L on the word-length of phrases
- a lower bound M on their frequency
- eliminate phrases for which at least an ε fraction occurs on a single domain (produced by spammers.)

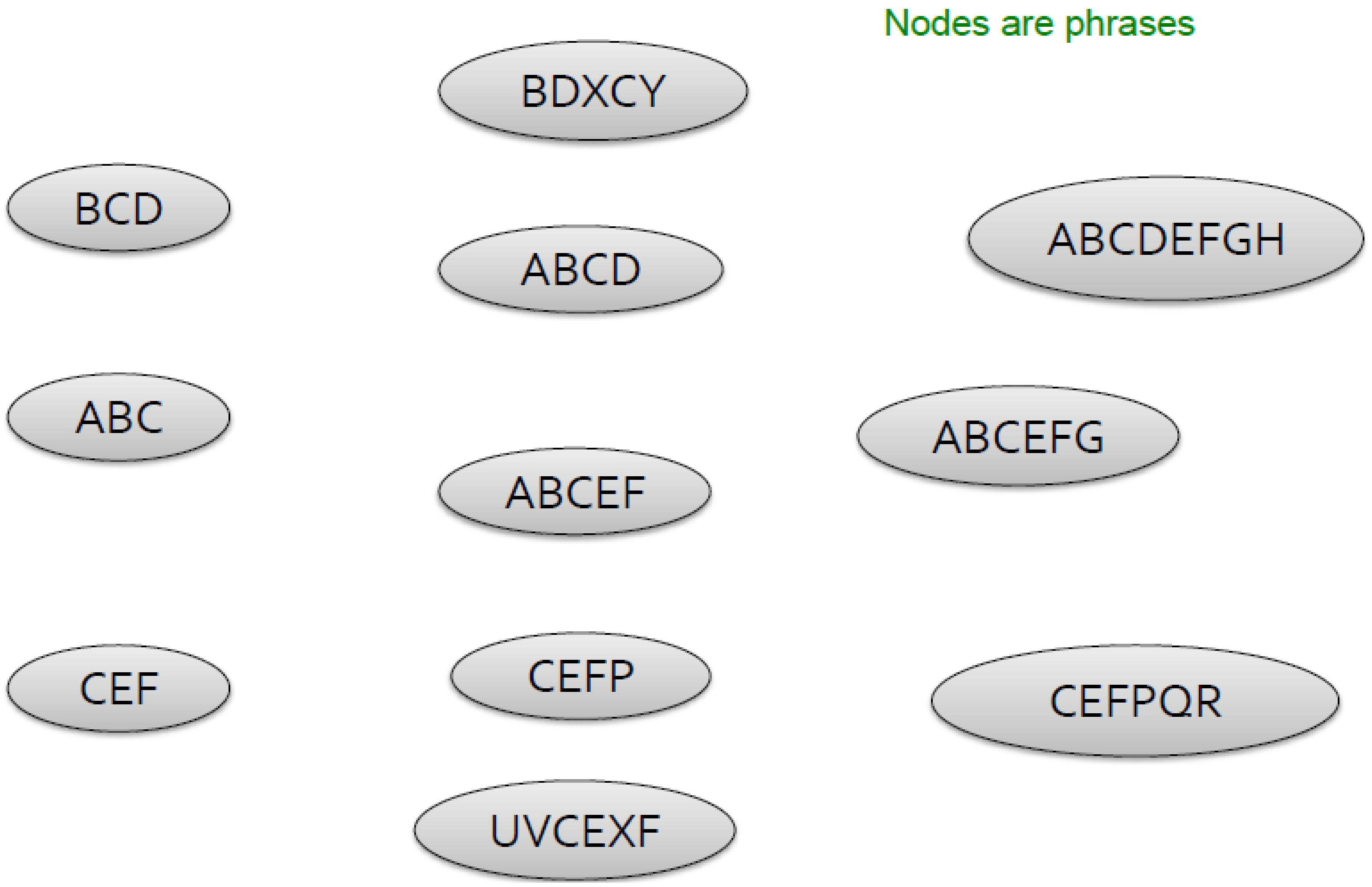
Phrase Graph Partitioning

Central idea: look for a collection of phrases that “belong” either to a single long phrase q, or to a single collection of phrases.

The outgoing paths from all phrases in the cluster should *flow into a single root node q* (node with no outgoing edges) -> look for a subgraph for which all paths terminate in a single root node.

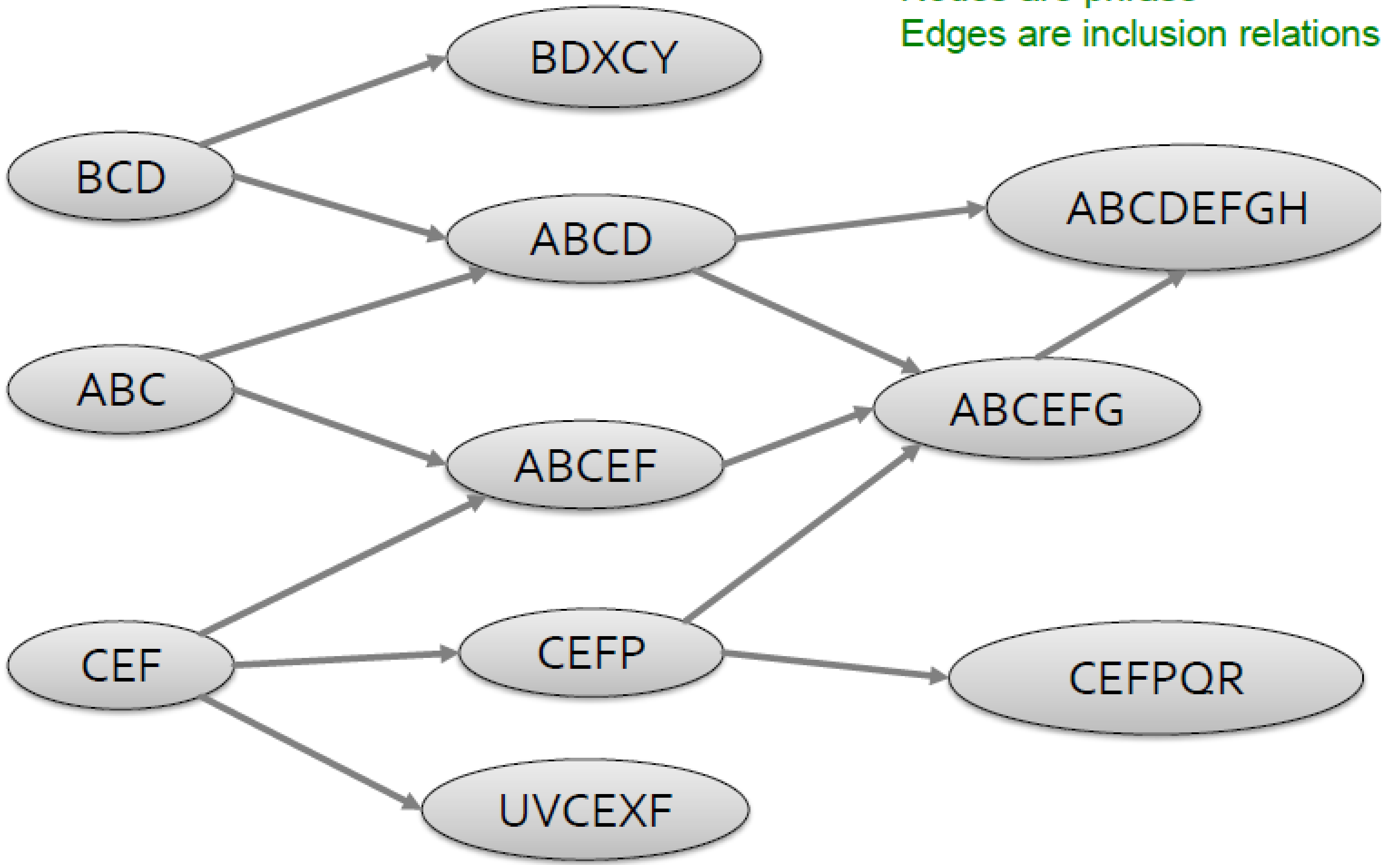
How?
Delete edges of small total weight from the phrase graph, so it falls apart into disjoint pieces, where each piece “feeds into” a single root phrase that can serve as the exemplar for the phrase cluster.

Phrase Graph



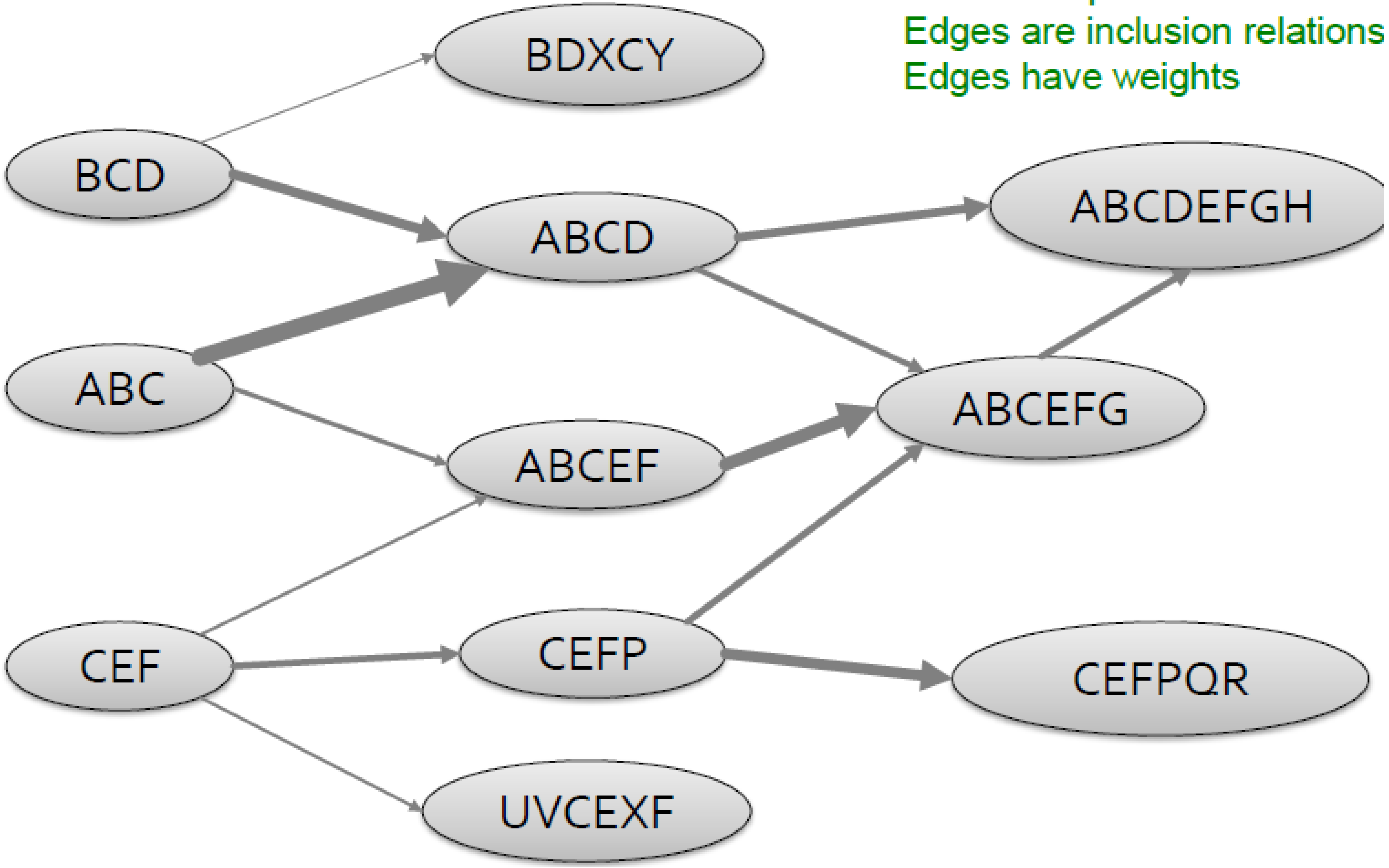
Phrase Graph

Nodes are phrase
Edges are inclusion relations



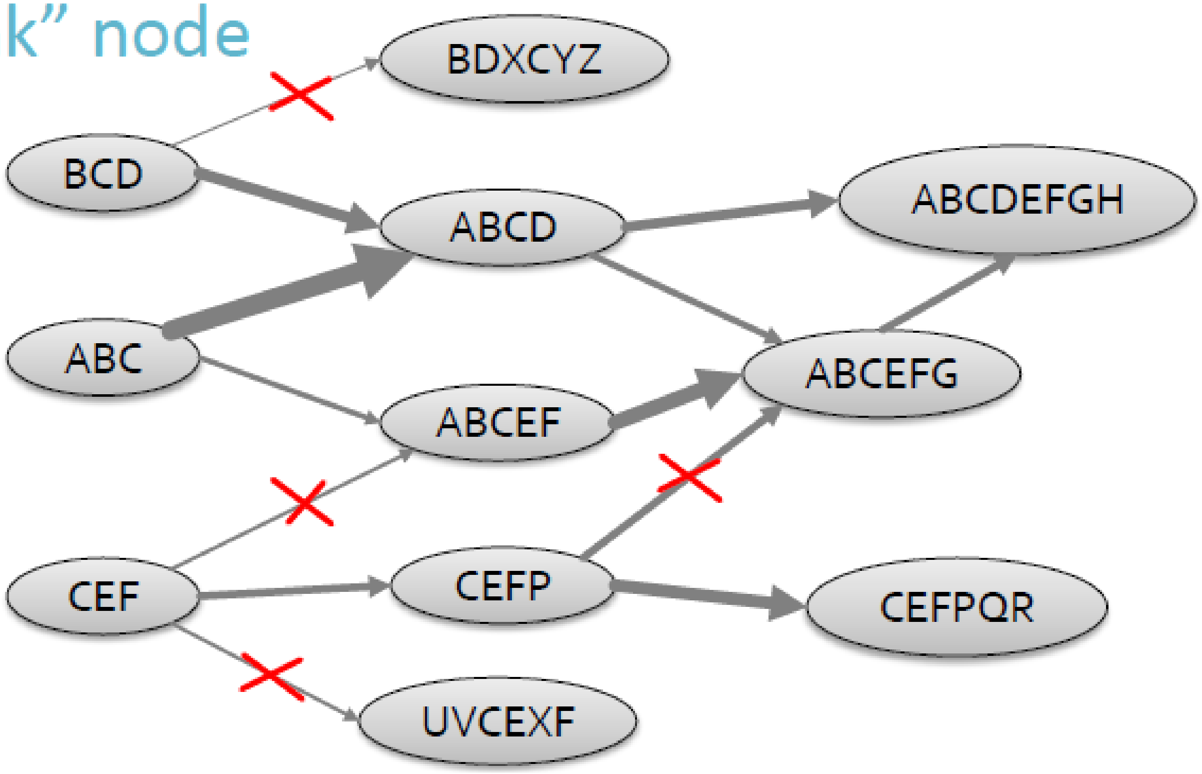
Phrase Graph

Nodes are phrases
Edges are inclusion relations
Edges have weights



Phrase Graph

Objective: In a directed acyclic graph (approx. phrase inclusion), delete min total edge weight s.t. each connected component has a single "sink" node



Phrase Graph Partitioning

The DAG Partitioning Problem: Given a directed acyclic graph with edge weights, delete a set of edges of minimum total weight so that each of the resulting components is single-rooted.

DAG Partitioning is NP-hard.

Phrase Graph Partitioning

In any optimal solution, there is at least one outgoing edge from each non-root node that has not been deleted.

A subgraph of the DAG where each non-root node has only a single out-edge must necessarily have single-rooted components, since the edge sets of the components will all be in-branching trees.

If **for each node** we happened to know **just a single edge** that was not deleted in the optimal solution, then the subgraph consisting of all these edges would have the same components (when viewed as node sets) as the components in the optimal solution of DAG Partitioning

It is enough to find a single edge out of each node that is included in the optimal solution to identify the optimal components.

Phrase Graph Partitioning Heuristic

Choose for each non-root node a single outgoing edge.

Which one?

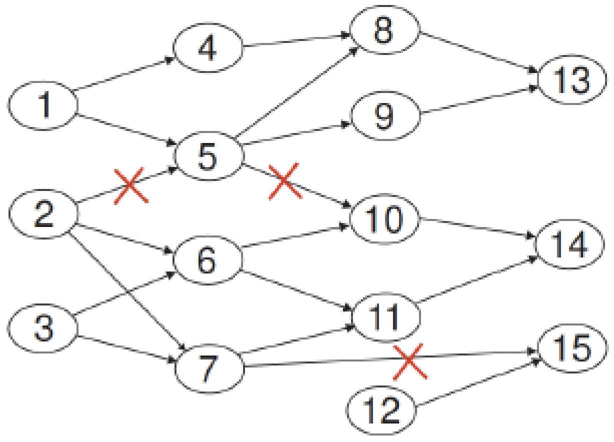
When compared to the total amount of edge weight kept in the clusters, if a random edge out of each phrase is kept.
an edge to the *shortest phrase* gives 9% improvement,
an edge to the *most frequent phrase* gives 12%

Proceed from the roots down the DAG and greedily assign each node to the cluster to which it has the most edges (gives 13% improvement)

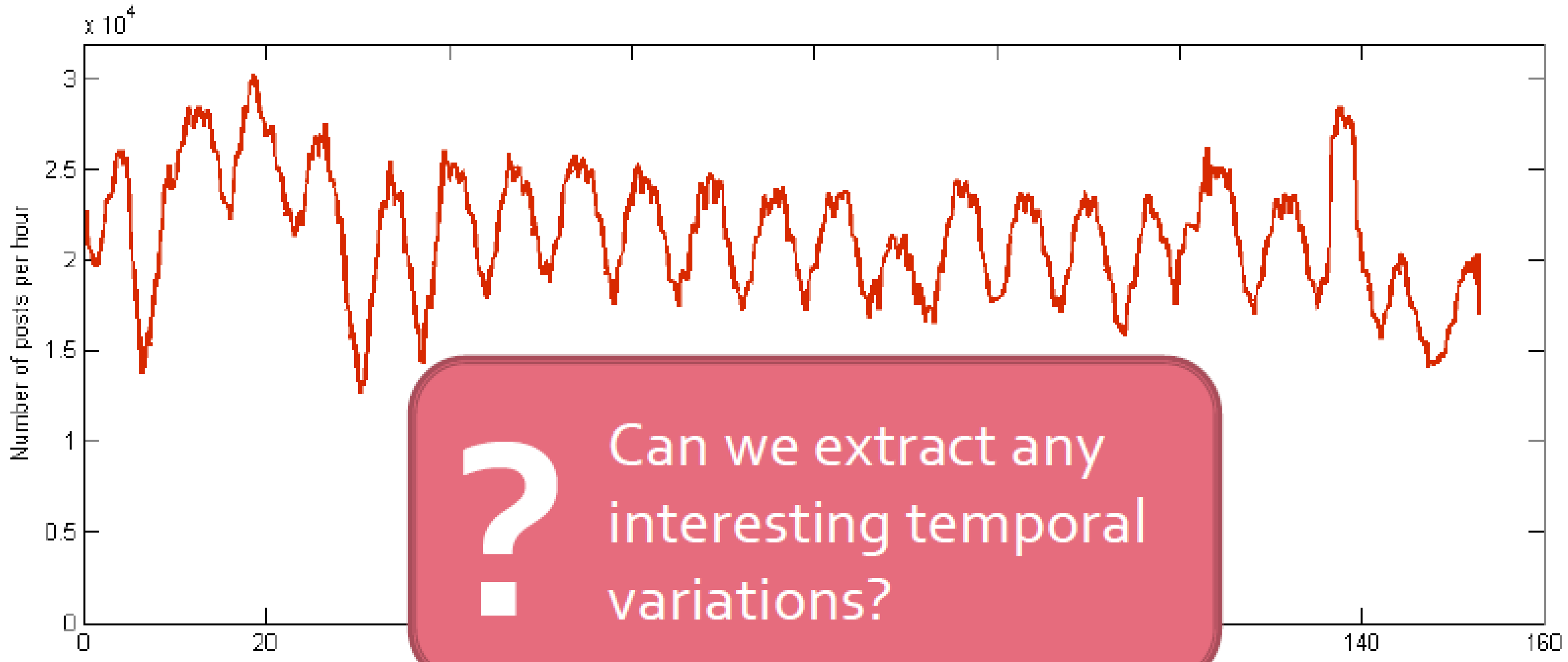
simulated annealing did not improve the solution

Data Set

- Since 2008 we have been collecting nearly all blog posts and news articles:
  - **6 billion documents**
  - **20 TB of data**
- **Solution:** Graph stream clustering
  - Phrases arrive in a stream
  - Simultaneously cluster the graph and attach phrases to the graph
  - Dynamically remove completed clusters



Temporal variation



? Can we extract any interesting temporal variations?

... is periodic, has no trends.

"Bandwidth" of the online media is constant

Temporal variation

Thread associated with a given phrase cluster: the set of all items (news articles or blog posts) containing some phrase from the cluster

Track all threads over time, considering both their individual temporal dynamics as well as their interactions with one another.

Temporal variation

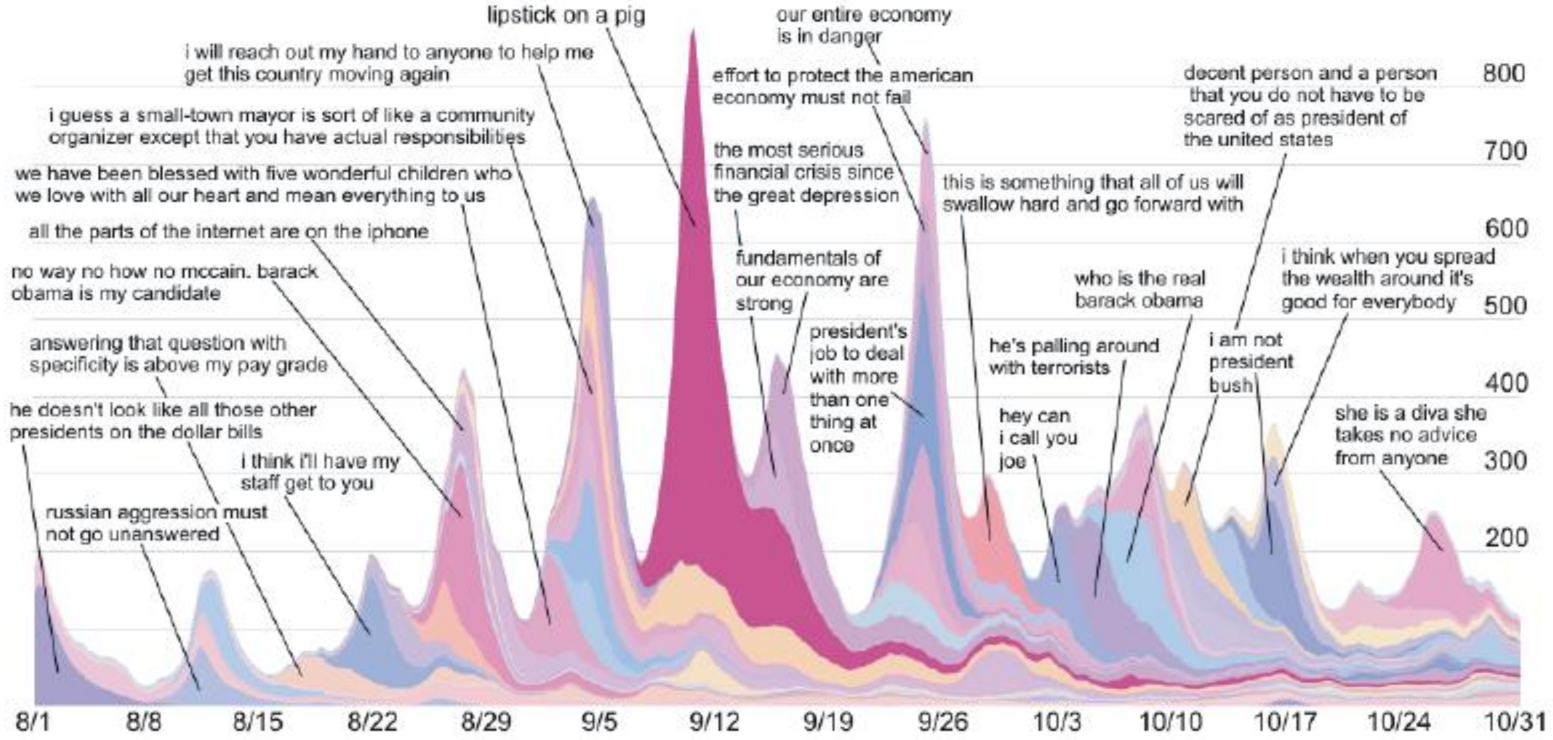


Figure 4: Top 50 threads in the news cycle with highest volume for the period Aug. 1 – Oct. 31, 2008. Each thread consists of all news articles and blog posts containing a textual variant of a particular quoted phrases. (Phrase variants for the two largest threads in each week are shown as labels pointing to the corresponding thread.) The data is drawn as a stacked plot in which the thickness of the strand corresponding to each thread indicates its volume over time. Interactive visualization is available at http://memetracker.org.

Thread volume increase and decay

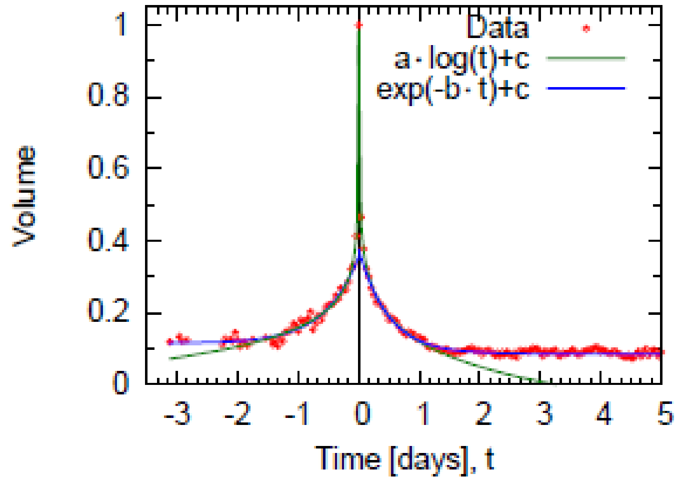


Figure 7: Thread volume increase and decay over time. Notice the symmetry, quicker decay than buildup, and lower baseline popularity after the peak.

Peak time of a thread: median time

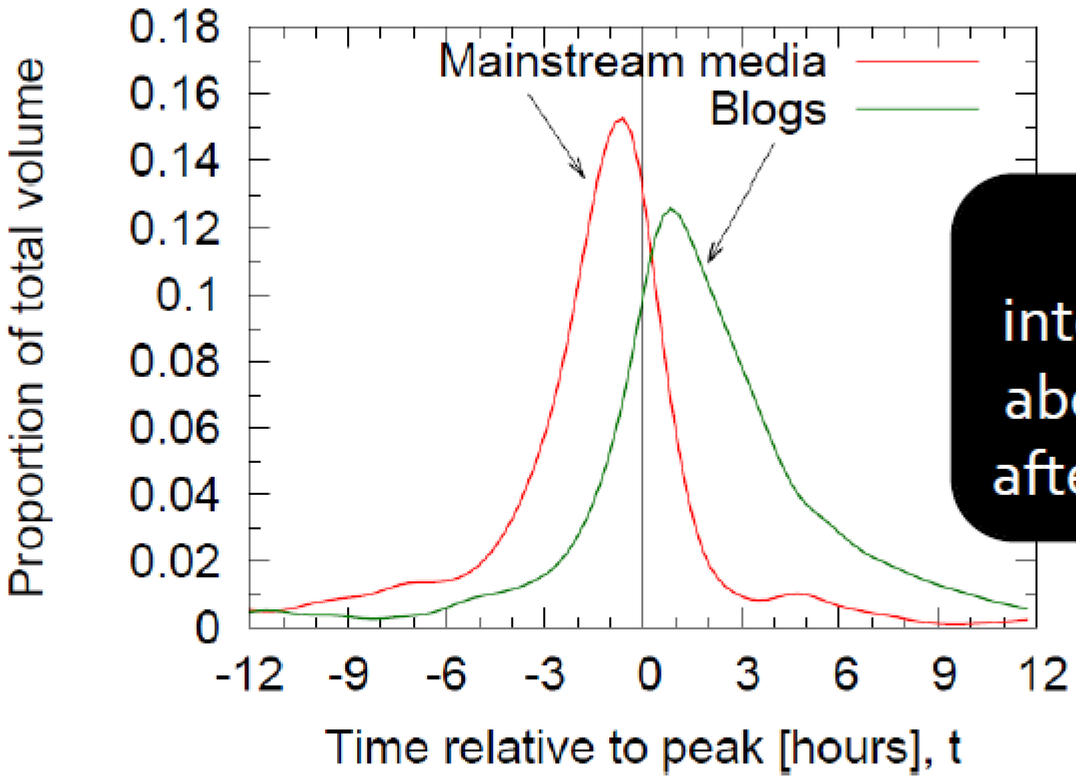
one would expect the overall volume of a thread to be very low initially; then the volume would rise; and slowly decay.

But rise and drop in volume surprisingly symmetric around the peak

Two distinct types of behavior:

- the volume outside an 8-hour window centered at the peak modeled by an exponential function
- the 8-hour time window around the peak is best modeled by a logarithmic function

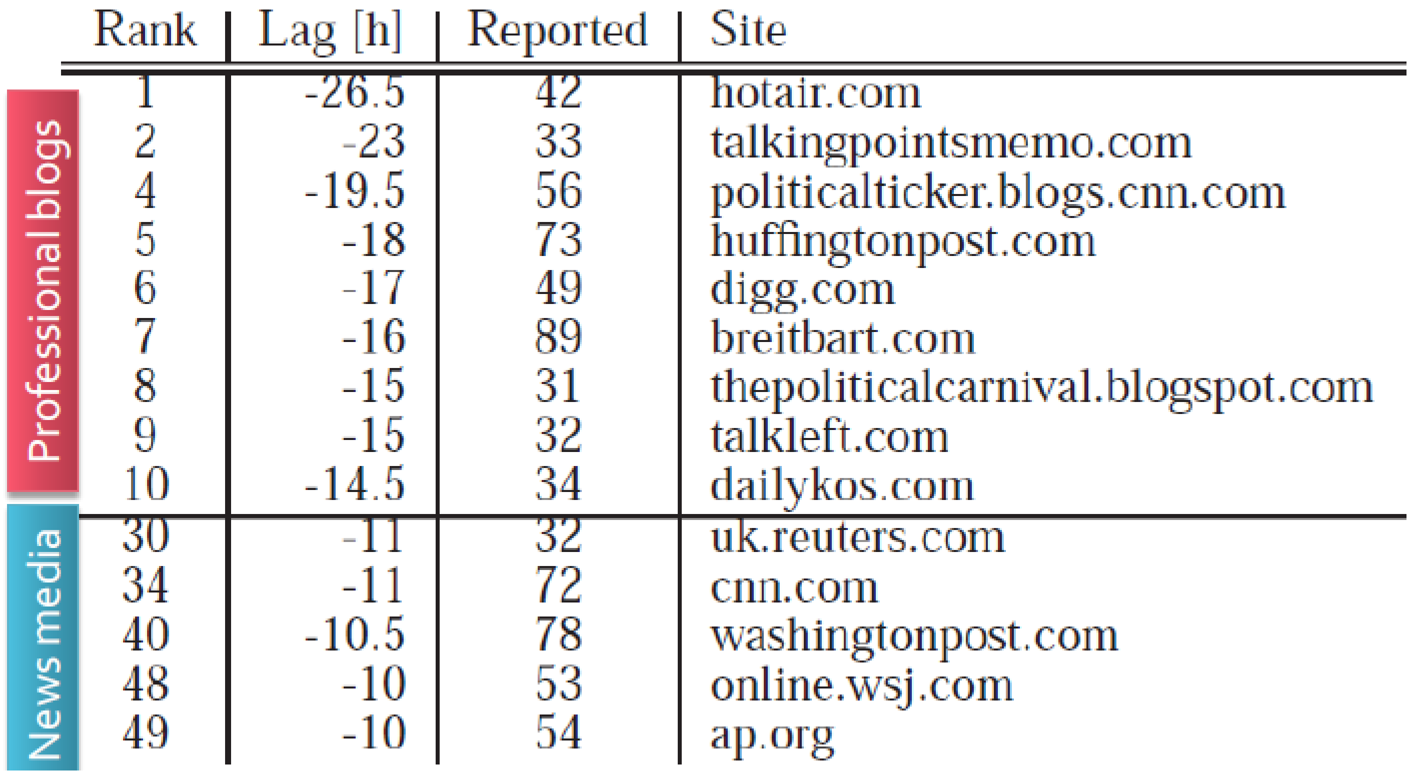
Lag between news and blogs



Peak blog intensity comes about 2.5 hours after news peak.

- **Using Google News we label:**
  - Mainstream media: 20,000 sites (44% vol.)
  - Blog (everything else): 1.6 million sites (56% vol.)

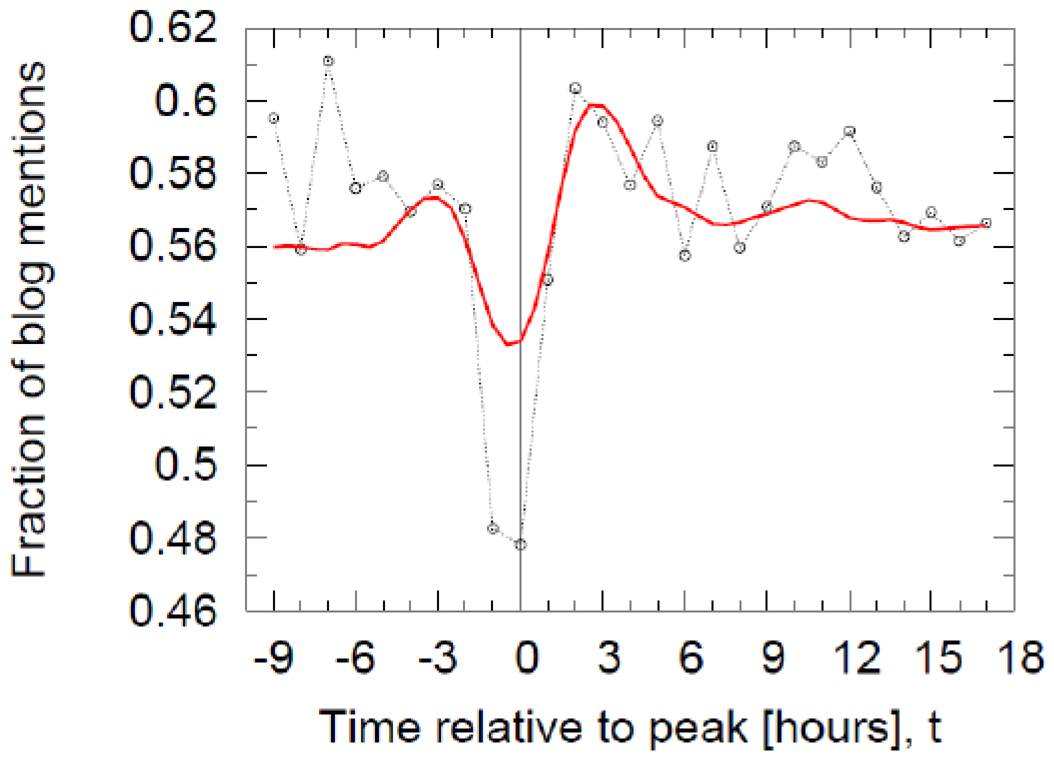
Lag of individual sites



	Rank	Lag [h]	Reported	Site
Professional blogs	1	-26.5	42	hotair.com
	2	-23	33	talkingpointsmemo.com
	4	-19.5	56	politicalticker.blogs.cnn.com
	5	-18	73	huffingtonpost.com
	6	-17	49	digg.com
	7	-16	89	breitbart.com
	8	-15	31	thepoliticalcarnival.blogspot.com
	9	-15	32	talkleft.com
	10	-14.5	34	dailykos.com
News media	30	-11	32	uk.reuters.com
	34	-11	72	cnn.com
	40	-10.5	78	washingtonpost.com
	48	-10	53	online.wsj.com
	49	-10	54	ap.org

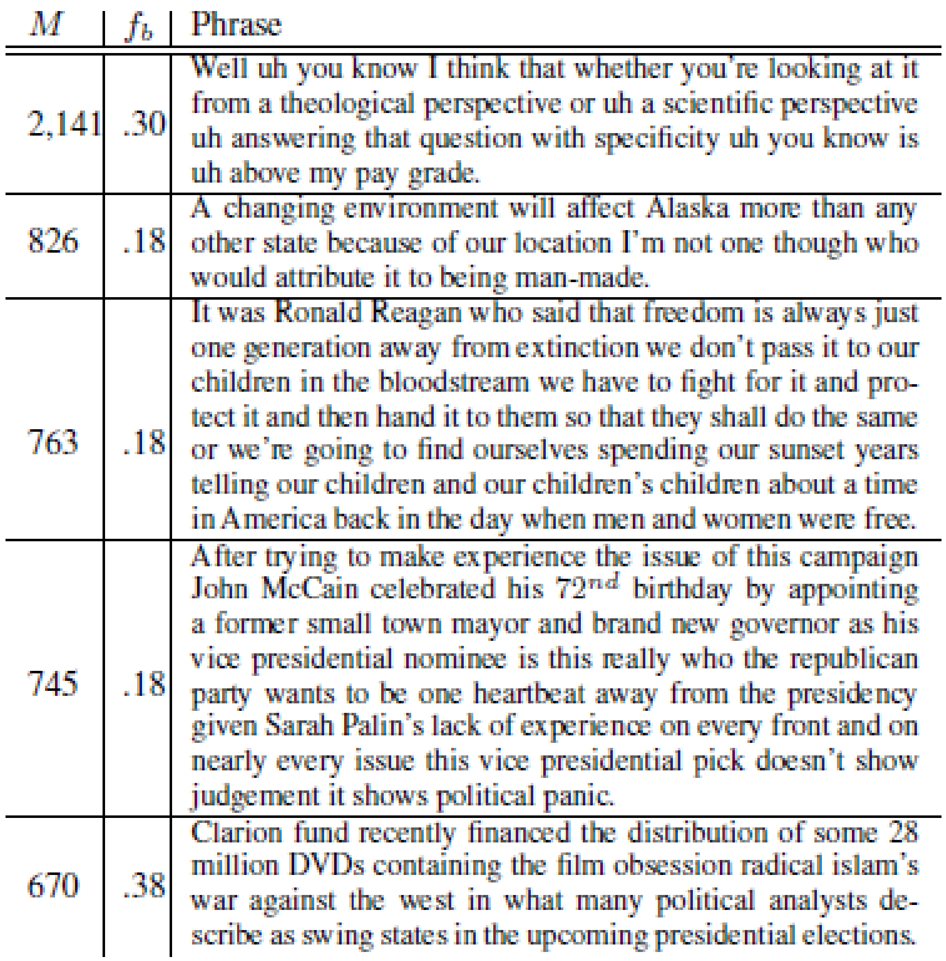
Oscillation of attention

ratio of blog volume to total volume for each thread as a function of time.



a “heartbeat”-like like dynamics where the phrase “oscillates” between blogs and mainstream media

Phrases discovered by blogs (3.5%)



M	f_b	Phrase
2,141	.30	Well uh you know I think that whether you're looking at it from a theological perspective or uh a scientific perspective uh answering that question with specificity uh you know is uh above my pay grade.
826	.18	A changing environment will affect Alaska more than any other state because of our location I'm not one though who would attribute it to being man-made.
763	.18	It was Ronald Reagan who said that freedom is always just one generation away from extinction we don't pass it to our children in the bloodstream we have to fight for it and protect it and then hand it to them so that they shall do the same or we're going to find ourselves spending our sunset years telling our children and our children's children about a time in America back in the day when men and women were free.
745	.18	After trying to make experience the issue of this campaign John McCain celebrated his 72nd birthday by appointing a former small town mayor and brand new governor as his vice presidential nominee is this really who the republican party wants to be one heartbeat away from the presidency given Sarah Palin's lack of experience on every front and on nearly every issue this vice presidential pick doesn't show judgement it shows political panic.
670	.38	Clarion fund recently financed the distribution of some 28 million DVDs containing the film obsession radical islam's war against the west in what many political analysts describe as swing states in the upcoming presidential elections.

Table 2: Phrases first discovered by blogs and only later adopted by the news media. M: total phrase volume, f_b: fraction of blog mentions before 1 week of the news media peak.

Conclusions

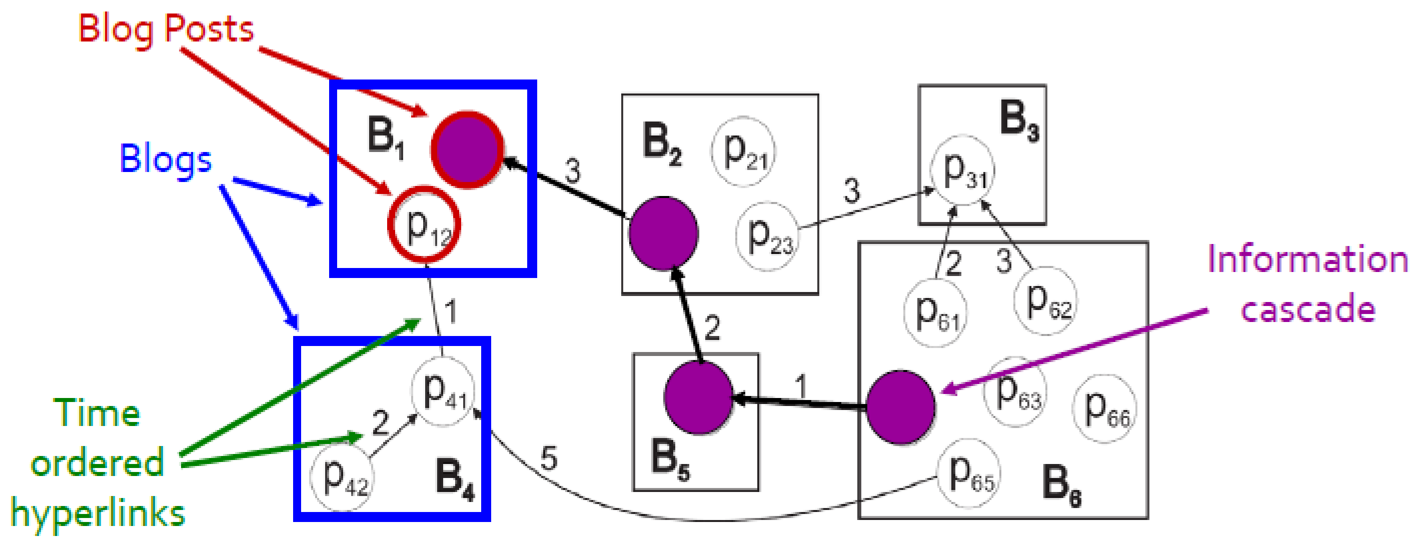
a framework for tracking short, distinctive phrases

scalable algorithms for identifying and clustering textual variants of such phrases that scale to a collection of 90 million articles

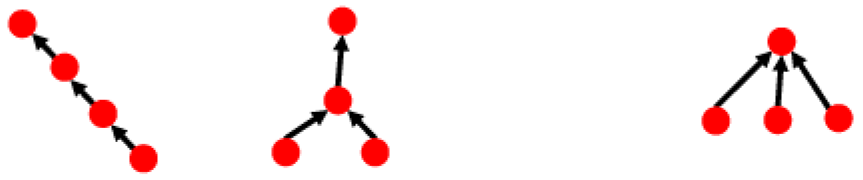
Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S. Glance, Matthew Hurst: ***Patterns of Cascading Behavior in Large Blog Graphs*.** SDM 2007

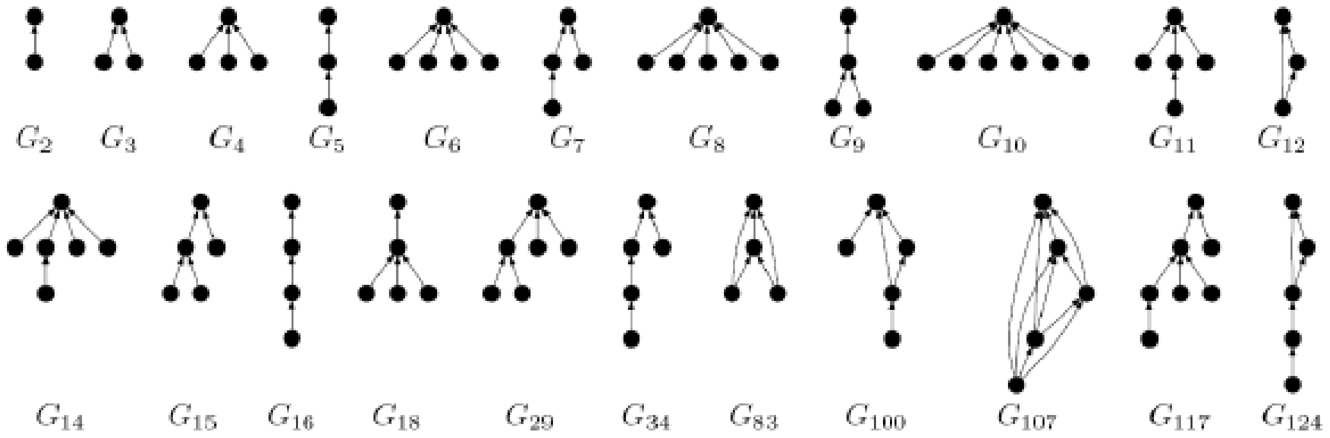
Slides are from Jure Leskovec's course
"On Social Information Network Analysis"

- **Tracking Hyperlinks on the Blogosphere**

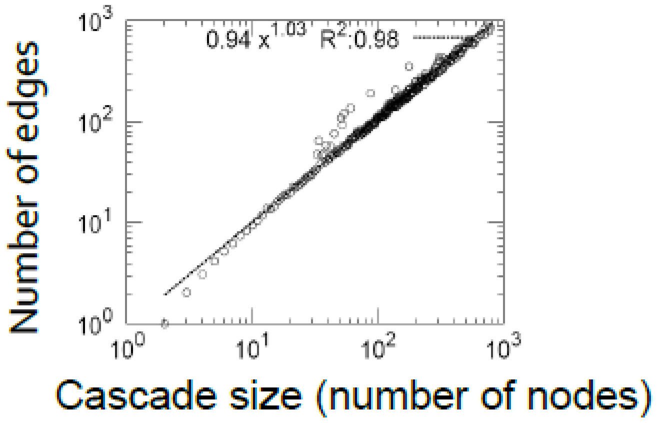


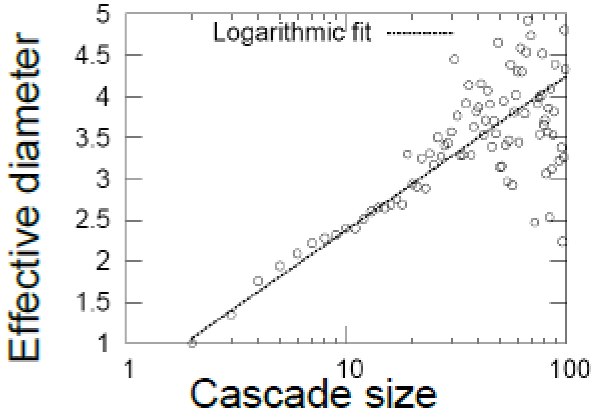
- **Identify cascades** – graphs induced by a time ordered propagation of hyperlinks





Cascade shapes (ranked by frequency)

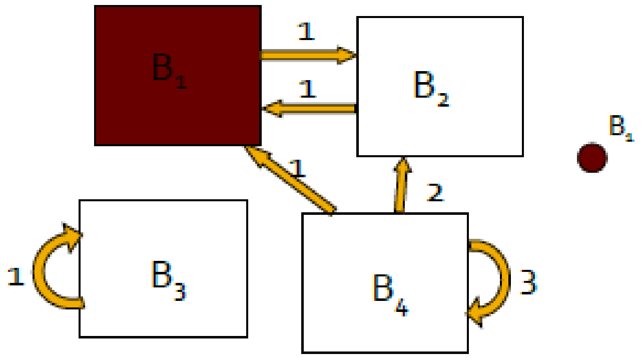




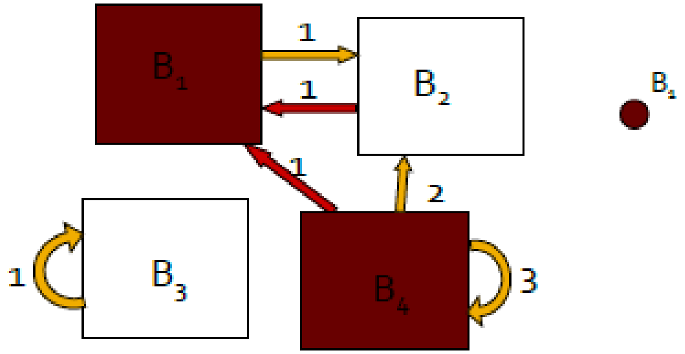
- **Most of cascades are trees:**
  - Number of edges is smaller than the number of nodes in a cascade
  - Diameter increases logarithmically

- Cascade sizes follow a heavy-tailed distribution
  - Viral marketing:
    - Books: steep drop-off: power-law exponent -5
    - DVDs: larger cascades: exponent -1.5
  - Blogs:
    - Power-law exponent -2
- What’s a good model?
  - What role does the underlying social network play?
  - Can make a step towards more realistic cascade generation (propagation) model?

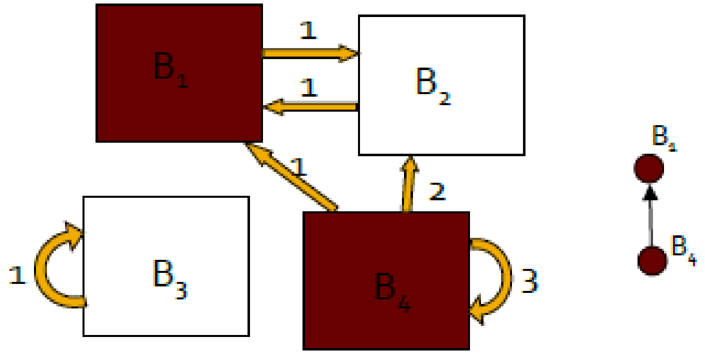
1) Randomly pick blog to infect, add to cascade.



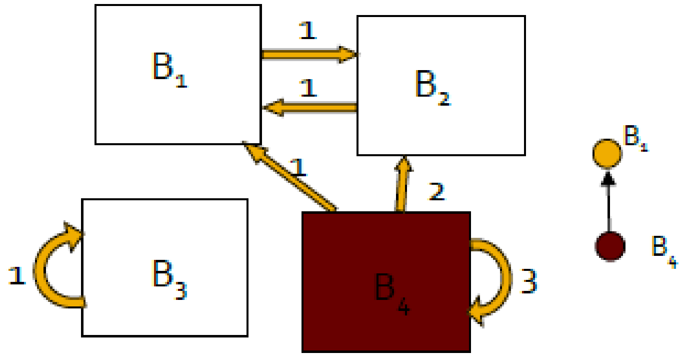
2) Infect each in-linked neighbor with probability β.



3) Add infected neighbors to cascade.

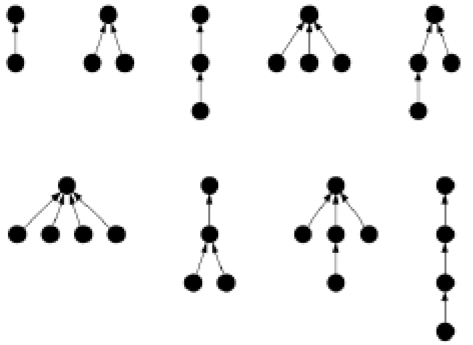


4) Set node infected in (i) to uninfected.

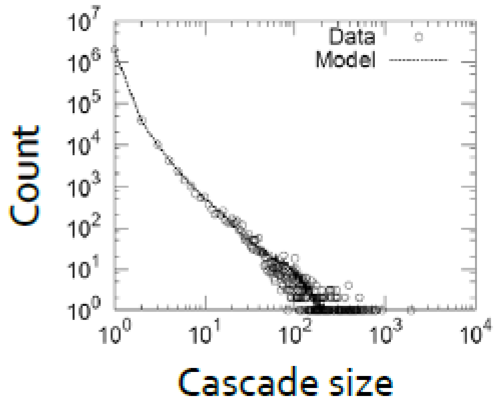


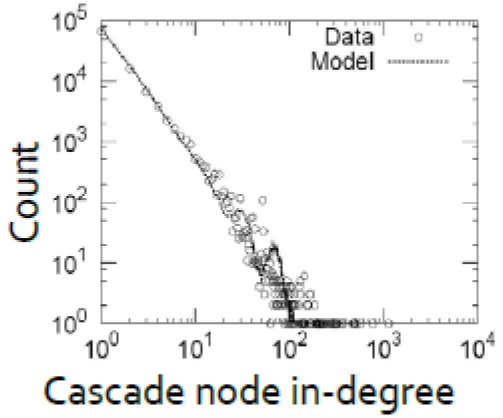
Generative model produces realistic cascades

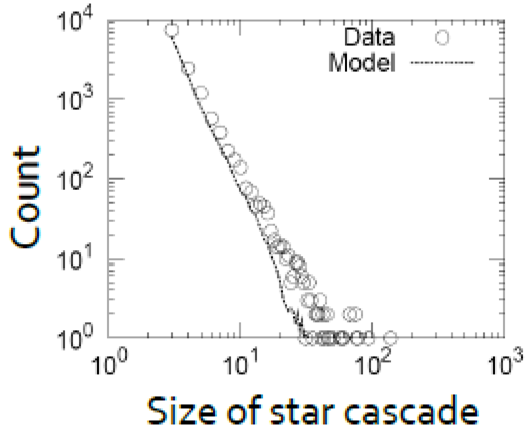
$\beta=0.025$

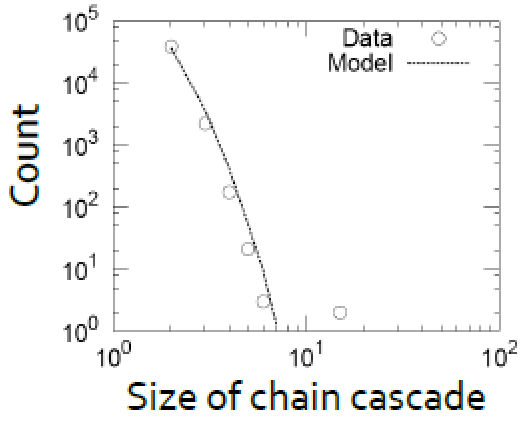


Most frequent cascades


Data
Model
Count
Cascade size


Data
Model
Count
Cascade node in-degree


Data
Model
Count
Size of star cascade


Data
Model
Count
Size of chain cascade