Words Over Time

Lecture 12: Text Analytics for Big Data
Mark Keane, Insight/CSI, UCD

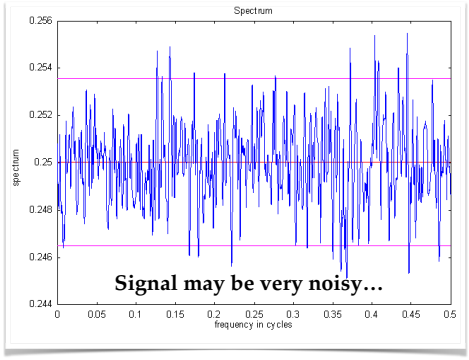
Time is Important

- Most of our lives occur over time; "Whatever is begotten, born, and dies"
- When such events are described in words, you go from word counts to how word-counts change over time
- So, we start to look trends, bursts, periodicity

Words in Time Eg

- Imagine, you want to see if a B-lister is becoming more popular
- You could sets up monitors on their name and look at how they change over time
- Several complexities can arise in this…

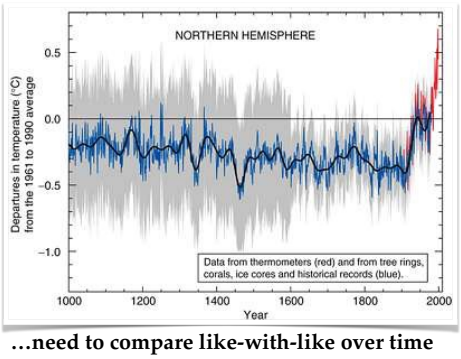
Issues: Noise



Signal may be very noisy…

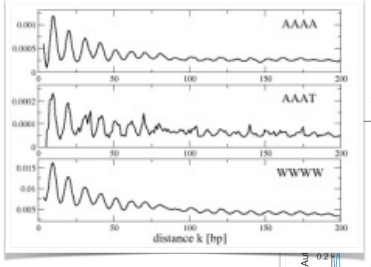
Issues: Normalisation

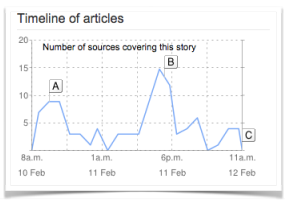
Raw frequencies may be misleading if not normalised…

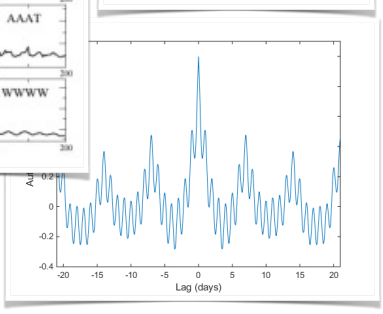


…need to compare like-with-like over time

Issues: Shapes







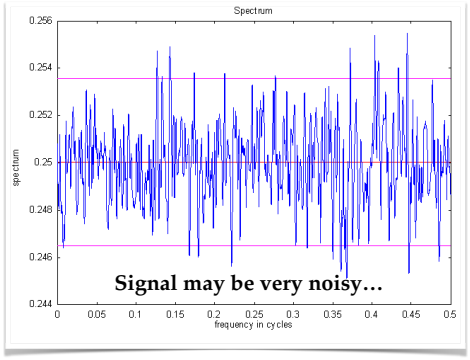
Trends & Periodicities may not be simple…

Basically It's…

- The idea of tracking events/people/opinions through changes in word occurrences over time deals with many of these issues
- Several different approaches:
  - Finding Exceptions over Time
  - Predicting Events from Word Trends
  - Predicting Happenings from Burstiness
  - Clustering Words from Shape Trends
  - Autocorrelation

Finding Exceptions in Time

Issues: Noise



Key Techniques

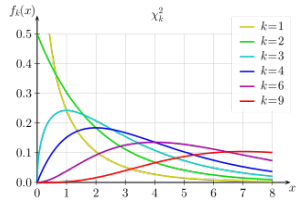
- Many traditional, non-parametric statistics are directed at assessing differences in frequencies
- X^2, G^2, Fisher's Exact Test all determine if nominal variables are independent/associated
- They do not assume normal distributions but may use binomial and other distributions

X^2: Test of Independence

Answer

Sex	Yes	No
Male (N=100)	50	50
Female (N=100)	10	90

probability density function



k is number of variables

X^2 is pronounced "chi-squared" after Greek letter

Test of independence [edit]

In this case, an "observation" consists of the values of two outcomes and the null hypothesis is that the occurrence of these outcomes is statistically independent. Each observation is allocated to one cell of a two-dimensional array of cells (called a contingency table) according to the values of the two outcomes. If there are r rows and c columns in the table, the "theoretical frequency" for a cell, given the hypothesis of independence, is

$$E_{i,j} = \frac{\left(\sum_{i=1}^{c} O_{i,n_c}\right) \cdot \left(\sum_{j=1}^{r} O_{n_r,j}\right)}{N},$$

where N is the total sample size (the sum of all cells in the table). With the term "frequencies" this page does not refer to already normalised values.

The value of the test-statistic is

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}.$$

Fitting the model of "independence" reduces the number of degrees of freedom by $p = r + c - 1$. The number of degrees of freedom is equal to the number of cells rc, minus the reduction in degrees of freedom, p, which reduces to $(r-1)(c-1)$.

For the test of independence, also known as the test of homogeneity, a chi-squared probability of less than or equal to 0.05 (or the chi-squared statistic being at or larger than the 0.05 critical point) is commonly interpreted by applied workers as justification for rejecting the null hypothesis that the row variable is independent of the column variable.[4] The alternative hypothesis corresponds to the variables having an association or relationship where the structure of this relationship is not specified.

*From Wikipedia (2015, Feb)

X^2

N = 200

df(1)= 3.84
@ p < .05

X^2 crit=38.09

Therefore, Sex and Answers are not independent, or depend on one another in some way…

	Yes	No
Male (N=100)	50 (30)	50 (70)
Female (N=100)	10 (30)	90 (70)

Another Example*…

*From Wikipedia (2015, Feb)

Fairness of dice [edit]

A 6-sided dice is thrown 60 times. The number of times it lands with 1, 2, 3, 4, 5 and 6 face up is 5, 8, 9, 8, 10 and 20, respectively. Is the dice biased, according to the Pearson's chi-squared test at a significance level of

- 95%, and
- 99%?

n is 6 as there are 6 possible outcomes, 1 to 6. The null hypothesis is that the dice is unbiased, hence each number is expected to occur the same number of times, in this case, $\frac{60}{n} = 10$. The outcomes can be tabulated as follows:

i	O_i	E_i	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
1	5	10	−5	25	2.5
2	8	10	−2	4	0.4
3	9	10	−1	1	0.1
4	8	10	−2	4	0.4
5	10	10	0	0	0
6	20	10	10	100	10
			Sum		13.4

1	2	3	4	5	6
5 (10)	8 (10)	9 (10)	8 (10)	10 (10)	20 (10)

The number of degrees of freedom is $n - 1 = 5$. The *Upper-tail critical values of chi-square distribution* table gives a critical value of 11.070 at 95% significance level:

Degrees of freedom	Probability less than the critical value				
	0.90	0.95	0.975	0.99	0.999
5	9.236	11.070	12.833	15.086	20.515

As the chi-squared statistic of 13.4 exceeds this critical value, we reject the null hypothesis and conclude that the die is biased at 95% significance level.

At 99% significance level, the critical value is 15.086. As the chi-squared statistic does not exceed it, we fail to reject the null hypothesis and thus conclude that there is insufficient evidence to show that the die is biased at 99% significance level.

NB Assumptions*

Assumptions [edit]

The chi-squared test, when used with the standard approximation that a chi-squared distribution is applicable, has the following assumptions:[citation needed]

- Simple random sample – The sample data is a random sampling from a fixed distribution or population where every collection of members of the population of the given sample size has an equal probability of selection. Variants of the test have been developed for complex samples, such as where the data is weighted. Other forms can be used such as purposive sampling[5]
- Sample size (whole table) – A sample with a sufficiently large size is assumed. If a chi squared test is conducted on a sample with a smaller size, then the chi squared test will yield an inaccurate inference. The researcher, by using chi squared test on small samples, might end up committing a Type II error.
- Expected cell count – Adequate expected cell counts. Some require 5 or more, and others require 10 or more. A common rule is 5 or more in all cells of a 2-by-2 table, and 5 or more in 80% of cells in larger tables, but no cells with zero expected count. When this assumption is not met, Yates's Correction is applied.
- Independence – The observations are always assumed to be independent of each other. This means chi-squared cannot be used to test correlated data (like matched pairs or panel data). In those cases you might want to turn to McNemar's test.

A test that relies on different assumptions is Fisher's exact test; if its assumption of fixed marginal distributions is met it is substantially more accurate in obtaining a significance level, especially with few observations. In the vast majority of applications this assumption will not be met, and Fisher's exact test will be over conservative and not have correct coverage.[6]

*From Wikipedia (2015, Feb)

Text Analysis Eg

- Swan & Jensen (2000) TimeMines; for two time-periods, *t1* and *t2*, the no-of-times we see/do-not-see word *x* in document set *y*

	See-X	Do-not-see-X	*total*
t1	**10**	**100**	**110**
t2	**200**	**20**	**220**
total	**210**	**120**	**N = 330**

- If X^2-crit is sign. diff. then we can say that *x* at *t2* is independent of *x* at *t1*

TimeMines: Constructing Timelines with Statistical Models of Word Usage

Russell Swan and David Jensen

2.2 Finding Significant Features

A disimple statistic for discrete events—the presence or absence of a specified feature—is the number of documents that both contain a feature and occur during a specified time interval. The model for the arrival of these features is a random process with an unknown binomial distribution. We assume that 1: the random processes generating the features are stationary, meaning that they do not vary over time, and 2: the random processes for any pair of features are independent. With this default model of word usage we define our interestingness function as the amount of deviation from our default model.

	f_0	$\overline{f_0}$
$t \in t_0$	a	b
$t \notin t_0$	c	d

If the process producing feature f_0 is stationary, then for an arbitrary time period t_0 the probability of seeing the feature is the same as the probability of seeing the feature at other time periods of equal duration. Specifically, looking at the number of documents within which we see f_0 during time t_0 (a in table), the number of documents where we do not see f_0 during t_0 (b in table), the number of documents containing f_0 when $t \notin t_0$ (c in table), and the number of documents not containing f_0 when $t \notin t_0$ (d in Table), gives a 2×2 contingency table.

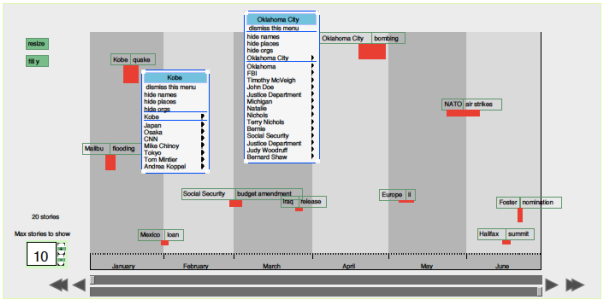
TimeMines Algorithm

- Compute word counts on each day, finding X^2 significant ones, then group contiguous days for a given word together and re-compute X^2 statistic for these new intervals
- Also, groups words together, in a given time period, by doing a X^2 on co-occurrence or not
- Displays events as groups of these words for particular time periods

TimeMines: Constructing Timelines with Statistical Models of Word Usage

Russell Swan and David Jensen

Department of Computer Science
University of Massachusetts
Amherst, Massachusetts USA



Overview of January - June, 1995. The topic labeled *Oklahoma City bombing* is the highest

References

Swan, R., & Allan, J. (2000, July). Automatic generation of overview timelines. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 49-56). ACM.

Swan, R., & Jensen, D. (2000, August). Timemines: Constructing timelines with statistical models of word usage. In *KDD-2000 Workshop on Text Mining* (pp. 73-80).

Other Techniques

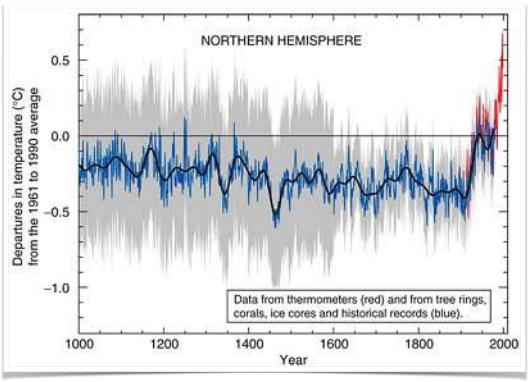
- There are several other X^2-like statistics that are often used for tests of independence and association; though they are all somewhat similar
- Stats: X^2, G^2,Fishers's Exact, McNemars tests
- See also: Kullbach-Liebler (KL), Expected Mutual Information Measure (EMIM), Pointwise Mutual Information, the Dice coefficient, Log-Likelihood-Ratios for topic signatures (LLR)

Some Caveats…

- REM, if your p = .05 and you test every day in a year, on chance alone, you will get 18 days where a difference is found
- So, issues about picking p-levels etc…
- See also: Moore, R. C. (2004, July). On Log-Likelihood-Ratios and the Significance of Rare Events. In *EMNLP* (pp. 333-340).

Using Words to Predict Change Over Time

Issues: Normalisation



Basically It's…

- Everyday events can be reflected in the momentum of words, in tweets, search queries, mails and so on
- Increasingly, text analytics can track these changes over time to predict movie revenue, disease spread, weather events and so on
- Techniques used are fairly simple

EG1. Asur & Huberman (2010)

- **Summary**: Predict movie box-office revenues from tweet chatter, better than trad. methods
- **Identify**: words that show movie in tweet
- **Plot**: use a simple tweet-rate measure
- **Predict**: simple linear model using tweet-rate

Asur, S., & Huberman, B. A. (2010, August). Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on* (Vol. 1, pp. 492-499). IEEE. NY.

Movies: Summary

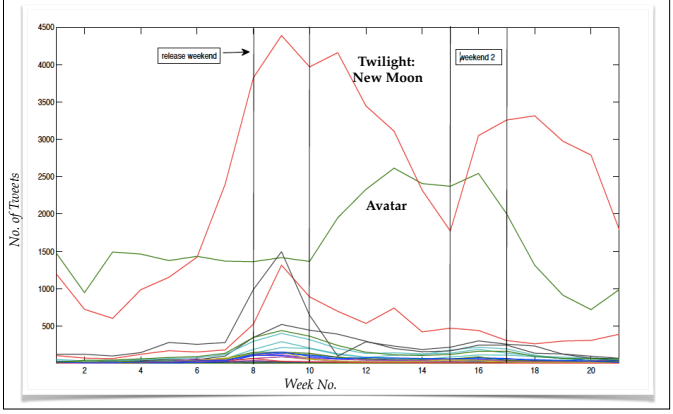
- People's discussions of brands and products in social media can be analysed for feedback, inform viral marketing and pre-release hype
- Twitter is esp. useful to track commentary, if one can identify key words in tweets
- Aim is to be better than Hollywood Stock Exchange (www.hsx.com)

Movies: Identify

- **DataSet:** 2.89M tweets, 1.2M users, 3 months from 2009-10, involving 24 movies; tweets 1 wk before & 2 wks after release
- **Word-Id**: tweets id-ed by keywords, title, promotional urls, photos, etc (details poor?)
- **NB**: Urls-incl and RTs not greatly predictive

Movie	Release Date
Armored	2009-12-04
Avatar	2009-12-18
The Blind Side	2009-11-20
The Book of Eli	2010-01-15
Daybreakers	2010-01-08
Dear John	2010-02-05
Did You Hear About The Morgans	2009-12-18
Edge Of Darkness	2010-01-29
Extraordinary Measures	2010-01-22
From Paris With Love	2010-02-05
The Imaginarium of Dr Parnassus	2010-01-08
Invictus	2009-12-11
Leap Year	2010-01-08
Legion	2010-01-22
Twilight : New Moon	2009-11-20
Pirate Radio	2009-11-13
Princess And The Frog	2009-12-11
Sherlock Holmes	2009-12-25
Spy Next Door	2010-01-15
The Crazies	2010-02-26
Tooth Fairy	2010-01-22
Transylmania	2009-12-04
When In Rome	2010-01-29
Youth In Revolt	2010-01-08

Movies: Shape of Attention



Movies: Plot

$$Tweet-rate(mov) = \frac{|tweets(mov)|}{|Time\ (in\ hours)|}$$

- Tweet-rate, no of tweets referring to a particular movie per hour (1 variable)
- For 24 movies, average tweet-rate in pre-release period correlates strongly-positively with gross box office take (r = .90; R2 = .80, p < 0.001)
- On 1st Weekend: Lowest was *Transylmania* @ $263,941 with 2.75 tweets per hour; Highest *Twilight* @ $142M with 1,365 tweets per hour

Movies: Predict

Features	Adjusted R^2	p-value
Avg Tweet-rate	0.80	3.65e-09
Tweet-rate timeseries	0.93	5.279e-09
Tweet-rate timeseries + thent	0.973	9.14e-12
HSX timeseries + thent	0.965	1.030e-10

TABLE IV
COEFFICIENT OF DETERMINATION (R^2) VALUES USING DIFFERENT PREDICTORS FOR MOVIE BOX-OFFICE REVENUE FOR THE FIRST WEEKEND.

- Linear regression of the time-series values of tweet-rates (7 days pre-release), plus *theatre-no* in which movie was released (8 vars)
- Very accurate predictor of box-office

Eg1.Asur & Huberman (2000): Afters

- They also did a sentiment analysis too, lesser role than tweet-rate; did generalised model
- Also, tested prediction for any weekend

Generalised Model

data collected regarding the product over time, in the form of reviews, user comments and blogs. Collecting the data over time is important as it can measure the rate of chatter effectively. The data can then be used to fit a linear regression model using least squares. The parameters of the model include:

- A : rate of attention seeking
- P : polarity of sentiments and reviews
- D : distribution parameter

Let y denote the revenue to be predicted and ϵ the error. The linear regression model can be expressed as :

$$y = \beta_a * A + \beta_p * P + \beta_d * D + \epsilon \quad (4)$$

where the β values correspond to the regression coefficients. The attention parameter captures the buzz around the product in social media. In this article, we showed how the rate of tweets on Twitter can capture attention on movies accurately.

EG2. Ginsberg et al (2009): GFT

- **Summary**: Google Flu Trends (GFT) analyses location-sensitive search queries for flu-related terms to predict the flu outbreaks
- **Identify**: 45 statistically-relevant flu terms
- **Plot**: day-by-day changes search term numbers
- **Predict**: uses linear model

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature, 457(7232)*, 1012-1014.

Flu: Summary

- Flu causes 0.25-0.5M deaths p.a.; When people feel sick they google drugs, treatment, symptoms; Google "knows" all this info; see *GoogleTrends* and www.google.org/flutrends/ for current measures
- Certain queries are highly correlated with GP visits; Centre for Disease Control (CDC) monitors GP visits for Influenza Like Illnesses (ILIs) throughout US
- GFT builds an early-warning system for flu epidemics; but breaks during 2009 H1N1 epidemic, Why?

Flu: Identify

We sought to develop a simple model that estimates the probability that a random physician visit in a particular region is related to an ILI; this is equivalent to the percentage of ILI-related physician visits. A single explanatory variable was used: the probability that a random search query submitted from the same region is ILI-related, as determined by an automated method described below. We fit a linear model using the log-odds of an ILI physician visit and the log-odds of an ILI-related search query: $\text{logit}(I(t)) = \alpha\text{logit}(Q(t)) + \varepsilon$, where $I(t)$ is the percentage of ILI physician visits, $Q(t)$ is the ILI-related query fraction at time t, α is the multiplicative coefficient, and ε is the error term. $\text{logit}(p)$ is simply $\ln(p/(1-p))$.

- **DataSet:** 100sB queries from Google Logs for 5 years; selected ***50M*** most common queries in US; weekly counts of these queries by state; also had 5-years of ILI GP visits data
- **Normalisation**: *Q(t) is query fraction*; Divided the count for each query in a given week by the total number of online search queries submitted in that location during that week
- **Word-Id**: For every query-term, measure how well it predicts ILI data, in a given region, at a given time; got highest scoring terms using z-transformed correlations between 9 regions; combined set of *n* queries to maximise fit to ILI data (n = 45 best)

Flu: 45 key query terms

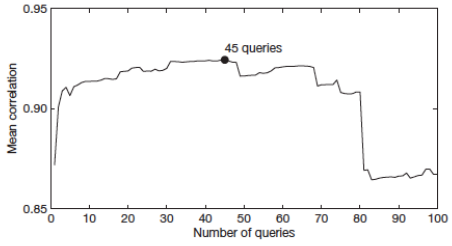


Figure 1 | An evaluation of how many top-scoring queries to include in the ILI-related query fraction. Maximal performance at estimating out-of-sample points during cross-validation was obtained by summing the top 45 search queries. A steep drop in model performance occurs after adding query 81, which is 'oscar nominations'.

Flu: 45 key query terms

Table 1 | Topics found in search queries which were found to be most correlated with CDC ILI data

Search query topic	Top 45 queries		Next 55 queries	
	n	Weighted	*n*	Weighted
Influenza complication	11	18.15	5	3.40
Cold/flu remedy	8	5.05	6	5.03
General influenza symptoms	5	2.60	1	0.07
Term for influenza	4	3.74	6	0.30
Specific influenza symptom	4	2.54	6	3.74
Symptoms of an influenza complication	4	2.21	2	0.92
Antibiotic medication	3	6.23	3	3.17
General influenza remedies	2	0.18	1	0.32
Symptoms of a related disease	2	1.66	2	0.77
Antiviral medication	1	0.39	1	0.74
Related disease	1	6.66	3	3.77
Unrelated to influenza	0	0.00	19	28.37
Total	45	49.40	55	50.60

The top 45 queries were used in our final model; the next 55 queries are presented for comparison purposes. The number of queries in each topic is indicated, as well as query-volume-weighted counts, reflecting the relative frequency of queries in each topic.

Flu: Plot

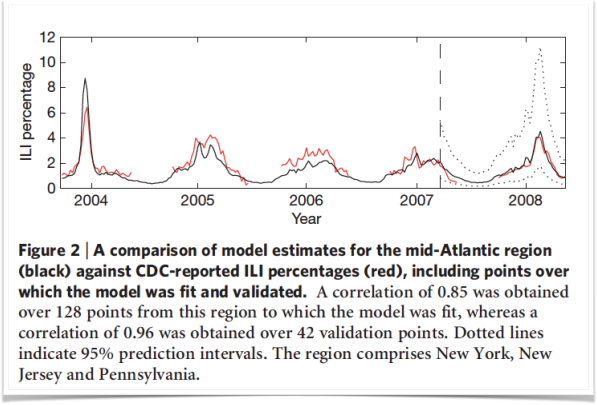


Figure 2 | A comparison of model estimates for the mid-Atlantic region (black) against CDC-reported ILI percentages (red), including points over which the model was fit and validated. A correlation of 0.85 was obtained over 128 points from this region to which the model was fit, whereas a correlation of 0.96 was obtained over 42 validation points. Dotted lines indicate 95% prediction intervals. The region comprises New York, New Jersey and Pennsylvania.

Flu: Predict

- Linear model with ILI-related query fraction as explanatory variable, was fitted to weekly ILI percentages between 2003-2007 for each of 9 US regions (mean $r = .90$)
- Validated against 42 held-out data points from the set (mean $r = .97$)
- Worked really well, at least until 2009

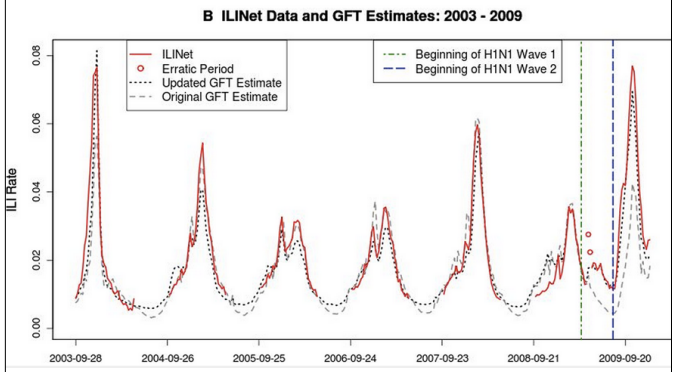
But(t)

"The search queries in our model are not, of course, exclusively submitted by users who are experiencing influenza-like symptoms, and the correlations we observe are only meaningful across large populations. Despite strong historical correlations, our system remains susceptible to false alerts caused by a sudden increase in ILI-related queries. An unusual event, such as a drug recall for a popular cold or flu remedy, could cause such a false alert."

Ginsberg et al (2009)

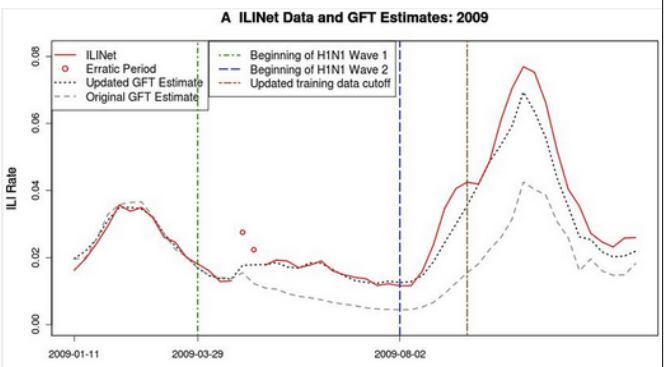
Then it Broke…

- From 2003-2009 GFT shows high correlation with ILI stats (ILINet) until 2009 H1N1 Pandemic (pH1N1)
- Then, it broke…because the search terms changed…



Cook, S., Conrad, C., Fowlkes, A. L., & Mohebbi, M. H. (2011). Assessing Google flu trends performance in the United States during the 2009 influenza virus A (H1N1) pandemic. *PloS one*, *6*, e23610.

The Break: Up close…



The Lesson

- GFT effectively has no real model of why users were using a particular search term
- The query terms were selected on purely statistical (and historical) grounds
- If querying-behaviour changes, then the predictive goodness of GFT can disappear

EG2. Flu: Afters

Methods Details

Constructing the ILI-related query fraction. We concluded the query selection process by choosing to keep the search queries whose models obtained the highest mean Z-transformed correlations across regions: these queries were deemed to be 'ILI-related'.

To combine the selected search queries into a single aggregate variable, we summed the query fractions on a regional basis, yielding our estimate of the ILI-related query fraction, $Q(t)$, in each region. Note that the same set of queries was selected for each region.

Fitting and validating a final model. We fit one final univariate model, used for making estimates in any region or state based on the ILI-related query fraction from that region or state. We regressed over 1,152 points, combining all 128 training points used in the query selection process from each of the nine regions. We validated the accuracy of this final model by measuring its performance on 42 additional weeks of previously untested data in each region, from the most recently available time period (18 March 2007 through to 11 May 2008). These 42 points represent approximately 25% of the total data available for the project, the first 75% of which was used for query selection and model fitting.

State-level model validation. To evaluate the accuracy of state-level ILI estimates generated using our final model, we compared our estimates against weekly ILI percentages provided by the state of Utah. Because the model was fit using regional data through 11 March 2007, we validated our Utah ILI estimates using 42 weeks of previously untested data, from the most recently available time period (18 March 2007 through to 11 May 2008).

METHODS

Automated query selection process. Using linear regression with fourfold cross validation, we fit models to four 96-point subsets of the 128 points in each region. Each per-query model was validated by measuring the correlation between the model's estimates for the 32 held-out points and the CDC's reported regional ILI percentage at those points. Temporal lags were considered, but ultimately not used in our modelling process.

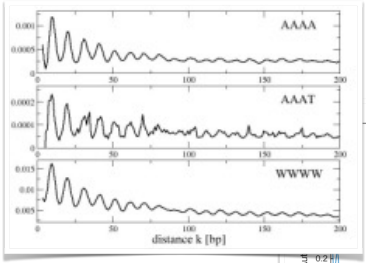
Each candidate search query was evaluated nine times, once per region, using the search data originating from a particular region to explain the ILI percentage in that region. With four cross-validation folds per region, we obtained 36 different correlations between the candidate model's estimates and the observed ILI percentages. To combine these into a single measure of the candidate query's performance, we applied the Fisher Z-transformation[11] to each correlation, and took the mean of the 36 Z-transformed correlations.

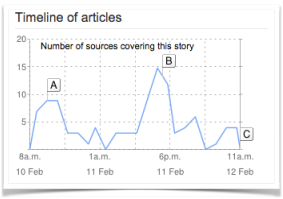
Computation and pre-filtering. In total, we fit 450 million different models to test each of the candidate queries. We used a distributed computing framework[12] to divide the work among hundreds of machines efficiently. The amount of computation required could have been reduced by making assumptions about which queries might be correlated with ILI. For example, we could have attempted to eliminate non-influenza-related queries before fitting any models. However, we were concerned that aggressive filtering might accidentally eliminate valuable data. Furthermore, if the highest-scoring queries seemed entirely unrelated to influenza, it would provide evidence that our query selection approach was invalid.

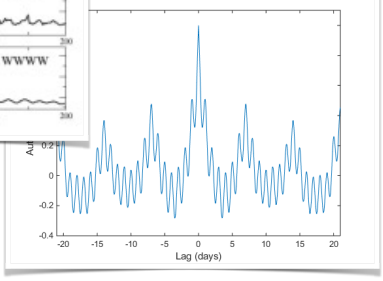
Ginsberg, J., et al. (2009). Detecting influenza epidemics using search engine query data. *Nature, 457(7232)*, 1012-1014.

Bursts

Issues: Shapes







Trends & Periodicities may not be simple…

Topics & Bursts

- Swan & Jensen looked at likelihood of a word in a document set, in a given time step, being higher than one would expect by chance
- Bursts deal with intervals of time between word-items; do those intervals occur at rates higher than one would expect by chance

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery, 7*, 373-397.

Texting Ones Love

- If you analysed the texts between two people, you might notice that even though they all involve the same keyword, there are characteristic bursts
- If you are about to meet, there may be a quick interchange of texts in a short period; whereas if you were arranging a party they may be spread over a week

Topics & Bursts

- Lots of things burst or have characteristic periodicities; topics in a document stream (eg blogs), trending hashtags, txts at a concert
- Kleinberg (2003) proposed a method to track such bursts; to classify items into groups, show extent in time, to build graphs of them

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery, 7*, 373-397.

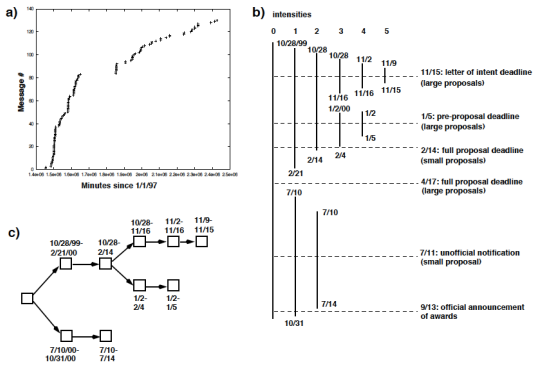


Figure 2: The stream of all messages containing the word "ITR," analyzed using the automaton $\mathcal{A}_2^*$. (a) The raw input data: the x-axis shows message arrival time; the y-axis shows message sequence number. (b) The set of bursts in the optimal state sequence for $\mathcal{A}_2^*$, drawn schematically to show the inclusions that form the tree Γ. (Lengths of intervals are standardized and hence not to scale.) Intervals are annotated with starting and ending dates, and the dates of the NSF ITR program deadlines are lined up with the intervals that contain them. (c) A representation of the tree Γ, showing inclusions among the bursts.

Kleinberg's Quote…

- "The appearance of a topic in a document stream is singled out by a 'burst of activity' with certain features rising sharply as the topic emerges"
- "Analysis of burst patterns may reveal latent hierarchical structure…that reflects content meaning in the stream"

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery, 7*, 373-397.

The Concept

- The algorithm groups gaps between items as belonging on one or more states (e.g., high-burst v low-burst state)
- States are captured in a Hidden Markov Model, with has a cost fn that penalises state-change, so that intervals are grouped into one state or another
- Algorithm delivers the optimal sequence of state groupings (based on cost) for overall sequence
- Also has weight for an item; a measure of its burstiness

How It Works…

- Kleinberg's algorithm analyses gaps between arrival times of text tokens are characterised as an exponential distribution
- Arrival times are modelled by a HMM model, that e.g. might have two states; high-rate state versus low-rate state
- Model is applied to actual arrivals of text tokens and probabilistically learn to group them by their gaps; tokens will group as high-rate or low-rate
- This can be expanded to any number gap groups based on different step changes (e.g. expect a geometric step change)
- HMM can have as many states as required for burst categories; and a cost function on transition between these different states (to sharpen clustering)

2.1 Kleinberg's Burst Model

Motivated originally by a problem of representing bursts of email messages, Kleinberg's burst algorithm [9] models bursts with an infinite state automaton in which each state represents a message arrival rate (of a Poisson arrival process). The higher the state, the smaller the expected time gap between messages. "Word bursts' can then be defined as having arrival rates defined by the number of messages containing a particular word. Additionally, jumping from a lower state to a higher state has an associated cost, while the cost to drop down from a higher state to a lower state is 0. Formally, these states are determined by optimization:

For a time series $\mathbf{x} = \{x_t \mid t = 0, 1, \ldots, n\}$ of *inter-arrival gaps*, find a state sequence $\mathbf{q} = \{q_t \mid t = 0, 1, \ldots, n\}$ minimizing cost function

$$c(\mathbf{q}|\mathbf{x}) = b(\mathbf{q}) \ln((1-p)/p) + \left(\sum_{t=0}^{n} -\ln f_{q_t}(x_t)\right)$$

where p is the probability of a state change, $b(\mathbf{q})$ is the number of state transitions (changes in successive states) in $\mathbf{q}$, and $f_i(x) = \alpha_i e^{-\alpha_i x}$ is the exponential density function for gap values x with arrival rate α_i.

By finding the optimal sequence of states minimizing the cost of transitions and the cost of differences between real arrival rate and the predicted emission rate, a time series of burst strengths is obtained. The complexity of the algorithm for finding this optimal sequence itself can be high, however; optimizing over the state space can be challenging for large-scale analyses or for online applications.

Kleinberg's recent paper on topic tracking in the news [7] formulates 'memes' as patterns of words, using the model of bursts developed earlier in [9]. A major contribution of the paper is to propose scalable clustering approaches for identifying short distinctive phrases traveling intact through on-line text.

Gory Details I

The sequence of states with the minimum cost can be calculated as follows. Let t be the index number of a document, $\tau(l, j)$ be a state transition cost from state l to state j, and $C_j(t)$ be a cost for document d_t and its state j.

1. For the initial state $t = 0$, define $C_0(t) = 0$, $C_1(t) = \infty$
2. $t = t + 1$
3. Calculate the cost $C_j(t)$ $(j = 0, 1)$
   $C_j(t)$ is defined as

$$C_j(t) = -\ln f_j(x_t) + \min_l (C_l(t-1) + \tau(l, j)) \quad (7)$$

   Here, the state of document d_t is described as j, and the state of d_{t-1} is described as l. In addition, the transition cost $\tau(l, j)$ is defined as $\tau(l, j) = \gamma \log n$ for $l < j$, and $\tau(l, j) = 0$ for $l >= j$.
4. Repeat steps 2 and 3 for all documents
5. Select the sequence of states that has the minimum cost
   The minimum cost sequence of states from calculated document-state pairs is determined using Viterbi algorithm. That is, starting with the state j of the document d_{last} with a minimum of $C_{j=0}(last)$ and $C_{j=1}(last)$, it recursively determines the state of the previous document by traversing the state transition from which state the current state comes.

Having determined the state of each document, continuous sequences of burst state documents are treated as bursts. Therefore, the length of a burst is defined as being from the arrival of the first burst document to the arrival of the last burst document.

Fujiki, T., Nanno, T., Suzuki, Y., & Okumura, M. (2004). Identification of bursts in a document stream. In *First International Workshop on Knowledge Discovery in Data Streams (in conjunction with ECML/PKDD 2004)* (pp. 55-64).

Gory Details II: Cost & Weight

The cost of a state sequence $\mathbf{q} = (q_{i_1}, \ldots, q_{i_n})$ in $\mathcal{B}^*_{s,\gamma}$ is defined as follows. If the automaton is in state i when the t^{th} batch arrives, a cost of

$$\sigma(i,t) = -\ln\left[\binom{d_t}{r_t} p_i^{r_t}(1-p_i)^{d_t - r_t}\right]$$

is incurred, since this is the negative logarithm of the probability that r_t relevant documents would be generated using a binomial distribution with probability p_i. There is also a cost of $\tau(i_t, i_{t+1})$ associated with the state transition from q_{i_t} to $q_{i_{t+1}}$, where this cost is defined precisely as for $\mathcal{A}^*_{s,\gamma}$. A state sequence of minimum total cost can then be computed as in Section 2.

type of LLR

prominence of this interval

Thus, the *two-state* automaton $\mathcal{B}^*_2$ is used; given an optimal state sequence, bursts of positive intensity correspond to intervals in which the state is q_1 rather than q_0. For such a burst $[t_1, t_2]$, we can define the *weight* of the burst to be

$$\sum_{t=t_1}^{t_2} (\sigma(1,t) - \sigma(0,t)).$$

In other words, the weight is equal to the improvement in cost incurred by using state 1 over the interval rather than state 0. Observe that in an optimal sequence, the weight of every burst is non-negative. Intuitively, then, bursts of larger weight correspond to more prominent periods of elevated activity. (This notion of weight can be naturally extended to

only change state if there is a big change in burst weight

E(e)gs & Techniques

- Burst algorithm has be used to show evolution of topic changes over time, the use of quote-memes in the news and other themes
  - Border & Mane (2004) evolution of topic areas in science
  - Kleinberg (2006) science docs
  - Earthquakes

Eg1. Borner & Mane (2004) Evolution of Science

- **Summary**: Analyses titles of PNAS papers to find topic trends over 20 years
- **Identify**: stated keywords and title words
- **Plot**: frequencies, burst, co-occurrences
- **Results**: shows science evolution

Mane, K. K., & Börner, K. (2004). Mapping topics and topic bursts in PNAS. *Proceedings of the National Academy of Sciences, 101(suppl 1)*, 5287-5290.

PNSA: Data

- **DataSet:** 47,073 PNSA papers, 1982-2001, selected 10% highly-cited papers (4,699) with 34,299 keywords (ISI, Medline MeSH) & titles
- **Word-Id**: computed frequencies and burst-weight for all 34k words; got intersection of highest 50 bursty-frequent words
- **NB**: also did term co-occurrence analysis; using a co-occurrence matrix (links in fig.)

PNSA: Plots: Frequencies

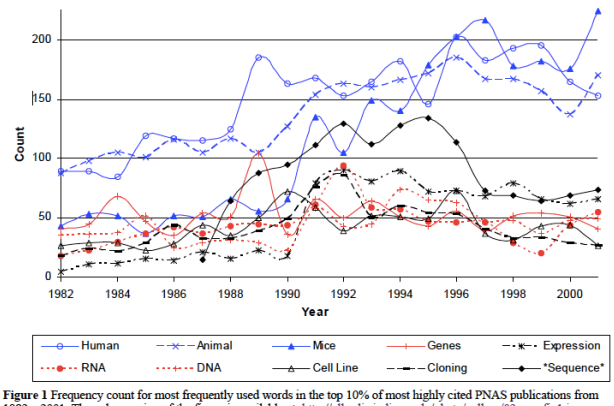
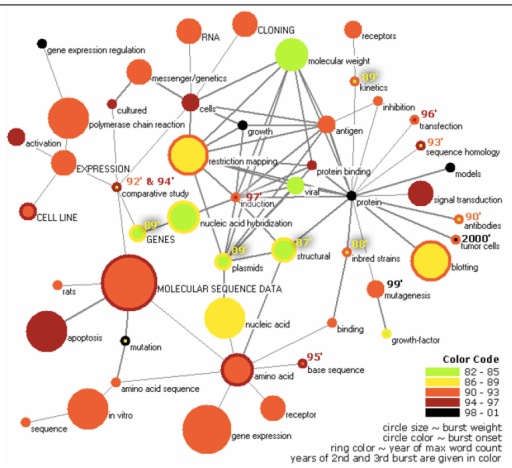


Figure 1 Frequency count for most frequently used words in the top 10% of most highly cited PNAS publications from 1982 – 2001. The color version of the figure is available at: http://ella.slis.indiana.edu/~katy/gallery/03pnas-fig1.jpg

PNSA: Plots



PNSA: Plots

Figure 2 Co-word space of the top 50 highly frequent and bursty words used in the top 10% most highly cited PNAS publications in 1982-2001. The color version of the figure is available at: http://ella.slis.indiana.edu/~katy/gallery/03pnas-

PNAS: Results

In the early 1980's the primary research foci were structural properties of biological entities such as cells, genes, etc. This was followed by a phase of research in kinetics and the study of the mutation behavior of genes. Toward the early 90's, in conjunction with the start of the human genome project, the research paradigm shifted toward sequence data studies. During this time period, molecular sequence data - amino acid sequences associated with the genome project - rose to prominence. Major funding via the human genome project also brought together several interconnected research areas primarily dealing with cloning, PCR, and gene expression depicting cloning studies. These experiments are an extension of prior studies on plasmids, genes,

Mane, K. K., & Börner, K. (2004). Mapping topics and topic bursts in PNAS. *Proceedings of the National Academy of Sciences, 101(suppl 1)*, 5287-5290.

Eg2. Kleinberg (2006)

- **Summary**: Similar to M&B, but in CS
- **Identify**: words in title of papers at SIGMOD and VLDB conferences, 1975-2001
- **Result**: burst weights of words show the periods in which particular terms dominate

Kleinberg, J. (2006). Temporal dynamics of on-line information streams. *Data stream management: Processing high-speed data streams.*

CS: Results

Word	Interval of burst
data	1975 SIGMOD — 1979 SIGMOD
base	1975 SIGMOD — 1981 VLDB
application	1975 SIGMOD — 1982 SIGMOD
bases	1975 SIGMOD — 1982 VLDB
design	1975 SIGMOD — 1985 VLDB
relational	1975 SIGMOD — 1989 VLDB
model	1975 SIGMOD — 1992 VLDB
large	1975 VLDB — 1977 VLDB
schema	1975 VLDB — 1980 VLDB
theory	1977 VLDB — 1984 SIGMOD
distributed	1977 VLDB — 1985 SIGMOD
data	1980 VLDB — 1981 VLDB
statistical	1981 VLDB — 1984 VLDB
database	1982 SIGMOD — 1987 VLDB
nested	1984 VLDB — 1991 VLDB
deductive	1985 VLDB — 1994 VLDB
transaction	1987 SIGMOD — 1992 SIGMOD
objects	1987 VLDB — 1992 SIGMOD
object-oriented	1987 SIGMOD — 1994 VLDB
parallel	1989 VLDB — 1996 VLDB
object	1990 SIGMOD — 1996 VLDB
mining	1995 VLDB —
server	1996 SIGMOD — 2000 VLDB
sql	1996 VLDB — 2000 VLDB
warehouse	1996 VLDB —
similarity	1997 SIGMOD —
approximate	1997 VLDB —
web	1998 SIGMOD —
indexing	1999 SIGMOD —
xml	1999 VLDB —

Fig. 1. The 30 bursts of highest weight using titles of all papers from the database conferences SIGMOD and VLDB, 1975-2001.

Eg3. Earthquakes (2010)

- Just in case you think all of this is too academic
- **Summary**: Early warning system using twitter to track earthquakes and typhoons in Japan
- **Identify**: uses classifier to identify keywords (about earthquakes and typhoons), notes location information
- **Plot**: mixture of a burst technique and a data fusion thing
- **Predict**: tracks the natural event

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web* (pp. 851-860). ACM.

EQuakes: Idea

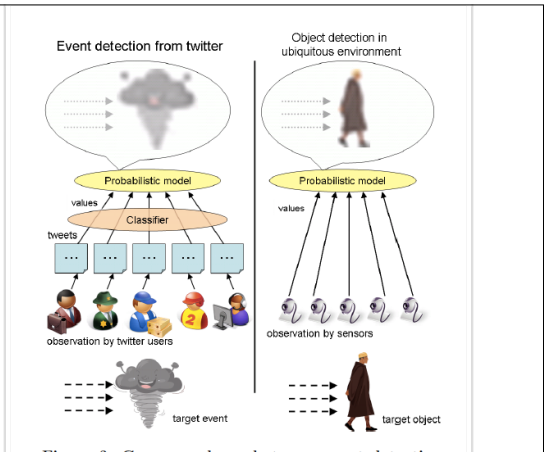


Figure 3: Correspondence between event detection from Twitter and object detection in a ubiquitous environment.

EQakes: Data

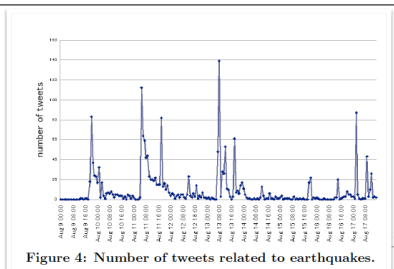


Figure 4: Number of tweets related to earthquakes.

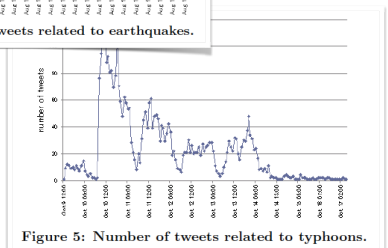


Figure 5: Number of tweets related to typhoons.

EQakes: Algorithm

Algorithm 2 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^{thre}_{occur}, 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.

EQakes: Results I

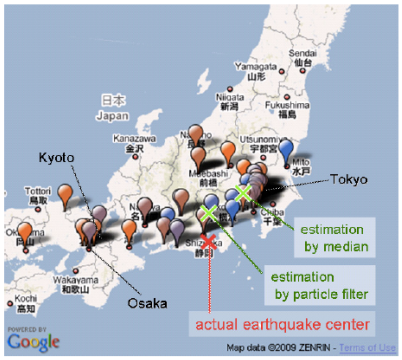


Figure 9: Earthquake location estimation based on tweets. Balloons show the tweets on the earthquake. The cross shows the earthquake center. Red represents early tweets; blue represents later tweets.

EQakes: Results II

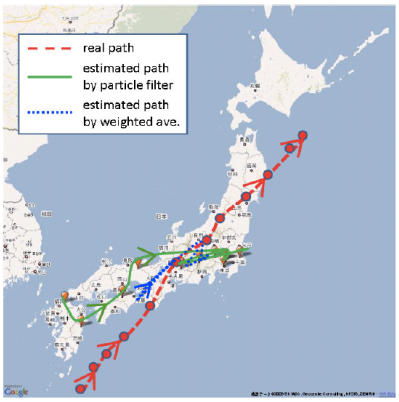
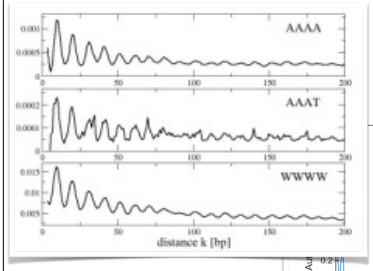
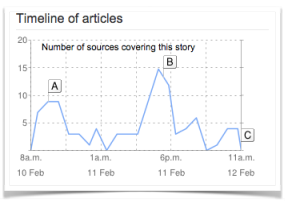


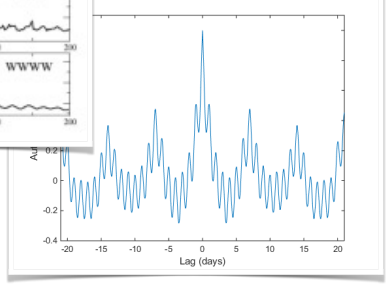
Figure 10: Typhoon trajectory estimation based on tweets.

Shape of Times Past: Messing With Graphs

Issues: Shapes







Trends & Periodicities may not be simple…

Shapes of Ideas

- Intuitively, we are seeing graphs with particular shapes, bursty peaks or repeated patterns
- Most of previous techniques do not address the shape issue, in and of itself
- Here we consider some quite different ones

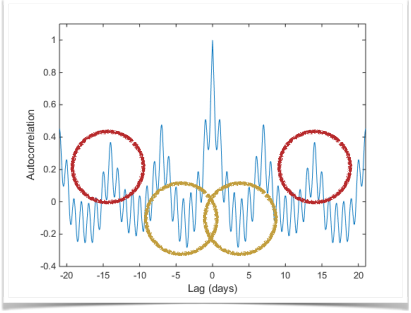
To Look at…

- Correlation and Autocorrelations
- Moving Averages & Smoothening
- (nb there are K-spectral methods for looking at similarity in graph shapes, though a minority activity…)

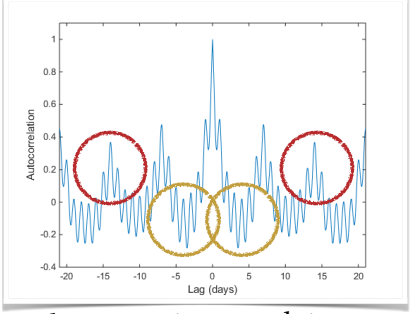
Stocks & Stockmarkets

- Stock markets are inherently about measure changes in values (price of a stock) over time
- Stock movements are really complex, they are impacted by many variables and factors that probably change over time (Fed Chair)
- Not surprisingly, a lot of methods have been developed to track stock movements to try to make predictions (eg, shape buying)

Autocorrelation



Correlation & Autocorrelation



Repeated patterns in a graph is to use autocorrelations between windowed portions

Autocorrelations

- *Definition*: autocorrelation or "lagged correlation" is a correlation of a time series with own past and future values (also called "persistence")
- Not surprisingly, a lot of methods have been developed to track stock movements to try to make predictions (eg, shape buying)

AutoC: The Idea

- Pearsons Correlation Coefficient (r) often represented by Greek letter *rho*
- In autocorrelation, we measure the covariance of two time-windowed samples, from the same data, divided by the square of their standard deviation

AutoC: Some Formulae

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Given measurements, $Y_1, Y_2, ..., Y_N$ at time $X_1, X_2, ..., X_N$, the lag k autocorrelation function is defined as

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}$$

Although the time variable, X, is not used in the formula for autocorrelation, the assumption is that the observations are equi-spaced.

Autocorrelation is a correlation coefficient. However, instead of correlation between two different variables, the correlation is between two values of the same variable at times X_i and X_{i+k}.

AutoC: Usage

- Often, you know what lag you want to check; but sometimes the AC will be computed for a whole range of lags to see which works
- Also remember, that you are comparing intervals with other intervals, so if you can to vary interval lengths too…this can be computationally expensive

Eg1. Predicting User Queries…

- **Summary**: How to predict different classes of user queries (+ url clicks); ones with rising trends/periodic repetition/once-off surprises
- **Identify**: temporal patterns in queries + clicks on websites; characterised as a limited set of predictive state-space models
- **System**: use classifier to identify features of queries-clicks that identify it as appropriate to one model or another; temporal/ domain-specific/periodicity features
- **Results**: Autocorrelation used to detect periodicity-features; performs well

Radinsky, K., Svore, K., Dumais, S., Teevan, J., Bocharov, A., & Horvitz, E. (2012). Modeling and predicting behavioral dynamics on the web.WWW-12 (pp. 599-608). ACM.

Eg1: The Idea

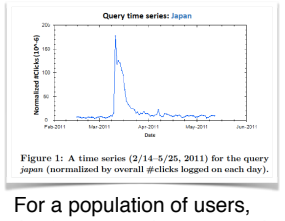


Figure 1: A time series (2/14–5/25, 2011) for the query *japan* (normalized by overall #clicks logged on each day).

For a population of users, the frequency that a query is issued and the number of times that a search result is clicked on for that query can change over time. We model such dynamics in the behavior as a time series, focusing on queries, URLs, and query-URL pairs as the behaviors. Quake 11/3/2011

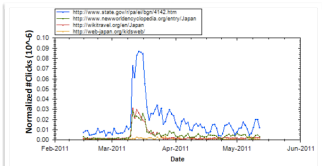


Figure 2: Time series (2/14–5/25, 2011) for sample clicked URLs for query *Japan* (normalized by total #clicks on each day).

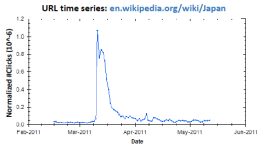


Figure 3: A Wikipedia article time series (2/14–5/25, 2011) for Japan (normalized by total #clicks on each day and position of the URL).

Some Are Hard to Predict…

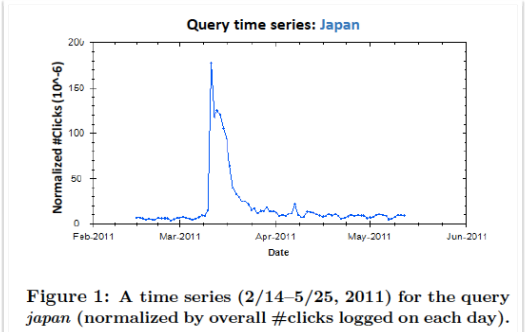


Figure 1: A time series (2/14–5/25, 2011) for the query *japan* (normalized by overall #clicks logged on each day).

Eg1: More Predictable…

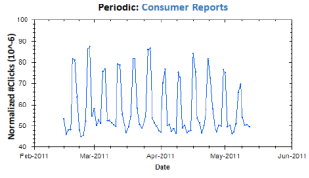


Figure 6: Query exhibiting a periodic behavior (normalized by overall #clicks on each day).

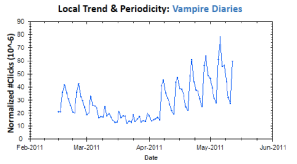


Figure 7: Query exhibiting periodic behavior with local trend (normalized by overall #clicks on each day).

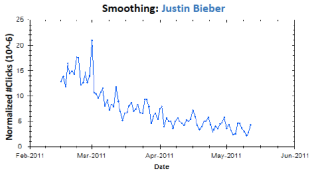


Figure 4: Query exhibiting behavior where historical

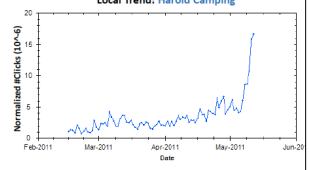


Figure 5: Query exhibiting behavior with local tre

Eg1: Detecting Periodicity

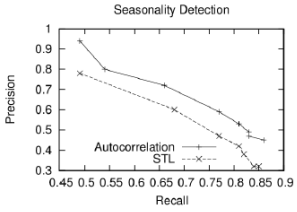


Figure 12: Comparison of STL method with Autocorrelation method for seasonality detection by precision (y axis) and recall (x axis).

Eg2. Predicting Stock Movements

- **Summary**: Complex study looking at Reuters news articles tool and changes in market measures (bid-offer spreads, volatility, trading volumes …) + sentiment
- **Identify**: arrival of relevant/irrelevant news items and the impact of same
- **Plot**: autocorrelations to capture periodic changes for particular variables
- **Results**: news-items have measured impacts on certain market changes

Groß-Klußmann, A., & Hautsch, N. (2011). When machines read the news. *Journal of Empirical Finance, 18*, 321-340.

Eg2: The Idea

This paper addresses the challenge of linking a virtually continuous and nonscheduled asset-specific news flow to intraday market activity. The fundamental objective of this study is to analyze to which extent high-frequency movements in returns, volatility and liquidity can be explained by the underlying mostly nonscheduled news arrivals during a day. To overcome the major difficulty of structuring and filtering news we employ the trading signals of an automated news engine. Such engines are technological innovations fueled by the algorithmic trading industry which computerize the interpretation of news based on linguistic pattern recognition techniques. The news engines are designed to provide signals on the meaning and the relevance of news items for future price movements as well as for future volatility and liquidity situations.

To our best knowledge, the present study is the first one systematically analyzing data from an automated news engine. We use the Reuters NewsScope Sentiment Engine which classifies firm-specific news according to positive, neutral and negative author sentiments based on linguistic pattern analysis of the respective news story. A further crucial feature of the engine is a numeric indicator classifying the relevance of news as well as a variable indicating the novelty thereof. Exploiting these numeric indicators of news sentiment, relevance and novelty we relate the firm-specific news to high-frequency returns, volatility, trading intensity, trade sizes, trade imbalances, spreads and market depth.

In specific, we aim to answer the following research questions:

(i) Are there significant and theory-consistent market reactions in high-frequency returns, volatility and liquidity to the intraday news flow?
(ii) Is trading on news-driven, machine-generated trading signals profitable?
(iii) Is the machine-indicated relevance of news empirically supported by corresponding market reactions?

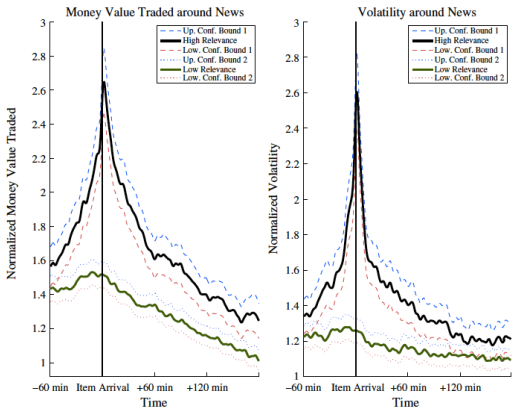


Fig. 3. Money value and volatility around news arrivals. Smoothed via kernel regression.

Moving Averages

Moving Averages…

- Moving Average (MA) or Rolling Average or Running Average
- *Definition*: a series of averages of different subsets of data points in a full data-set; here subsets of sliding time-windows
- There are many variants: simple, cumulative and weighed (using different schemes)

Moving Averages: The Idea

- Is simple: you smoothen the shape of the graph to make it more amenable to analysis
- Can remove noise that is hiding real trends
- Can reveal real periodic trends (G&K11)

(S)MA: The Formulae

example of a simple equally weighted running mean for a n-day sample of closing price is the mean of the previous *n* days' closing prices. If those prices are $p_M, p_{M-1}, \cdots, p_{M-(n-1)}$ then the formula is

$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

When calculating successive values, a new value comes into the sum and an old value drops out, meaning a full summation each time is unnecessary for this simple case,

$$SMA_{\text{today}} = SMA_{\text{yesterday}} + \frac{p_M}{n} - \frac{p_{M-n}}{n}$$

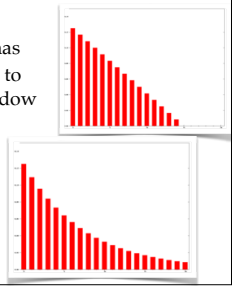
…for us, usually x items either side of some central value is taken as the window (for the n-day sample)

MA: Some Points

- Depending on window-size you may loose some data points at the beginning/end of the overall period
- Choice of window-size is an important design decision (G&K11-8 week was crucial); can still let thru' noise if it is shorter than the window-size
- You have choices about type of mean used; arithmetic or harmonic

Variants:

- *Cumulative MA (CMA)*: Average up until the current time point, re-compute cumulative value in next time step, updating *n* too.
- *Weighted MA (WMA)*: moving average that has multiplying factors to give different weights to data at different positions in the sample window (often decreasing arithmetic progression)
- *Exponential MA (EMA)*: weighting model is exponential; penalises older items, promotes newer items



ARMA-GARCH

- Time-series analysis is a mature area in Sciences and Economics: ARMA, GARCH, stationarity and so on…
- ARMA: Autoregressive–moving-average (ARMA) models provide a parsimonious description of a (weakly) stationary, stochastic process in terms of two polynomials, one for the auto-regression and the second for the moving average
- GARCH: If an autoregressive moving average model (ARMA model) is assumed for the error variance, the model is a generalized autoregressive conditional heteroskedasticity (GARCH), Bollerslev (1986)) model
- Often, you will see ARMA-GARCH models mentioned, where ARMA process handles autocorrelated price changes and the GARCH process handles volatility

Conclusions

- In this lecture, we have consider time-series data; where the data are various text-items
- This is increasingly more important as text analytics moves to consider streamed data over static corpora
- Note, very mature area of research in economics; really about importing it into TA; so it is more more conventional statistics (than e.g. ML)