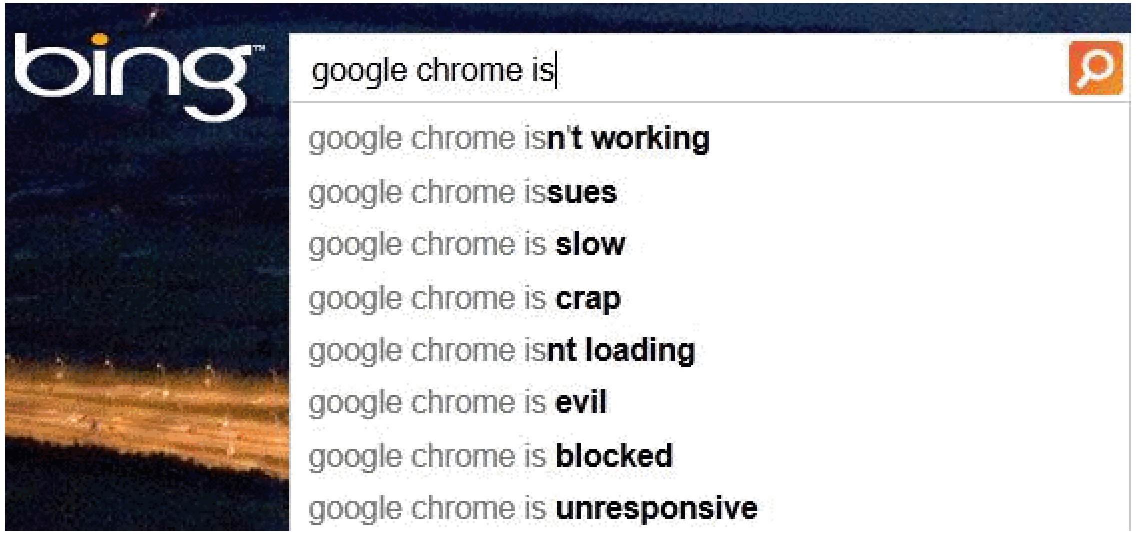
Instantaneous Estimation of Word Occurrences & Dynamic Language Modeling

Yuyan Wang, Ph.D. Candidate, Princeton University

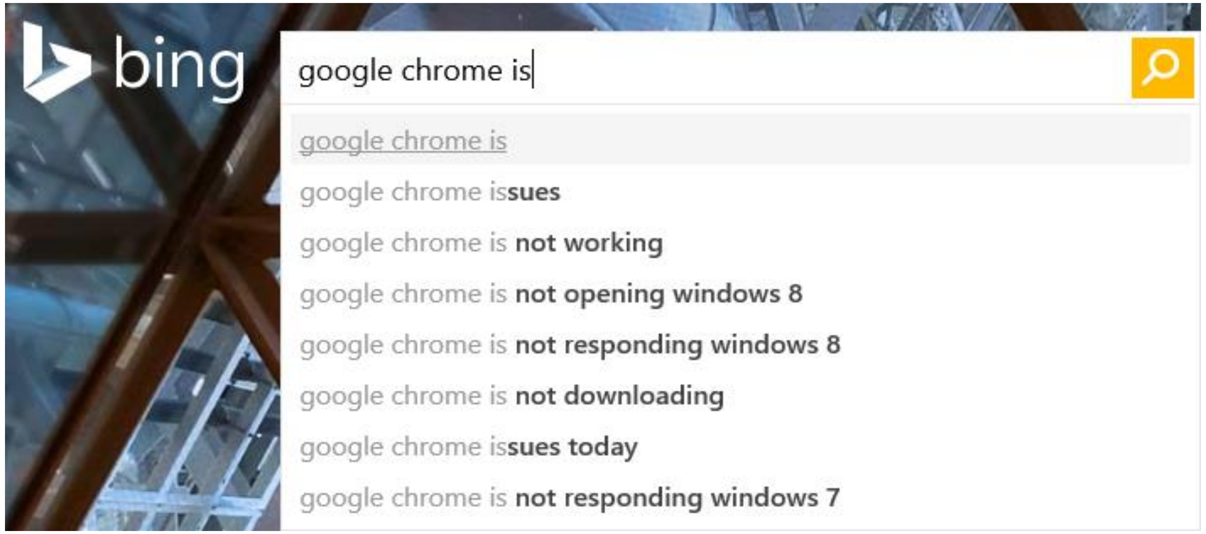
Paul Hsu, Senior Researcher, Microsoft Research

Part I:
Instantaneous Estimation of Word Occurrences

Background & Motivation: Online Queries



1 November, 2012



19 July, 2015

Fig.1: Popularity of queries change over time.

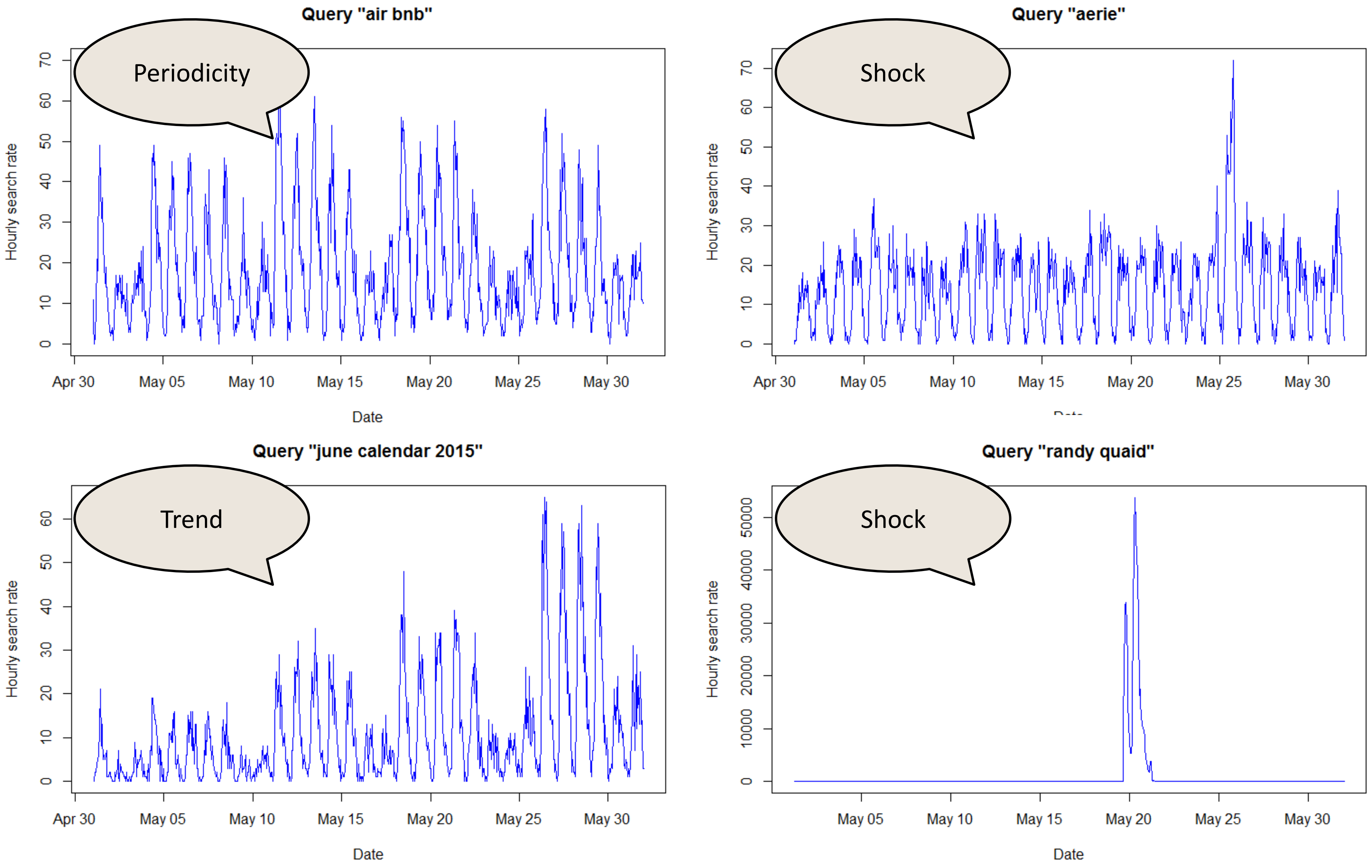


Fig.2: Different queries have different patterns: trend, periodicity, shock

Background & Motivation

- Query logs: Temporal info

  Features: periodicity, trend, shock…

- Data: a sequence of timestamps
- Goal: Predicting instantaneous query popularity (rate)

  Model **rates** instead of **prob.** of the words:

  e.g. Michael Jackson's death ⇨ Related queries ↑ ⇨ $P(\text{all other queries})$ ↓

- Potential Applications: query auto-completion; App store suggestion; Predictive Keyboard

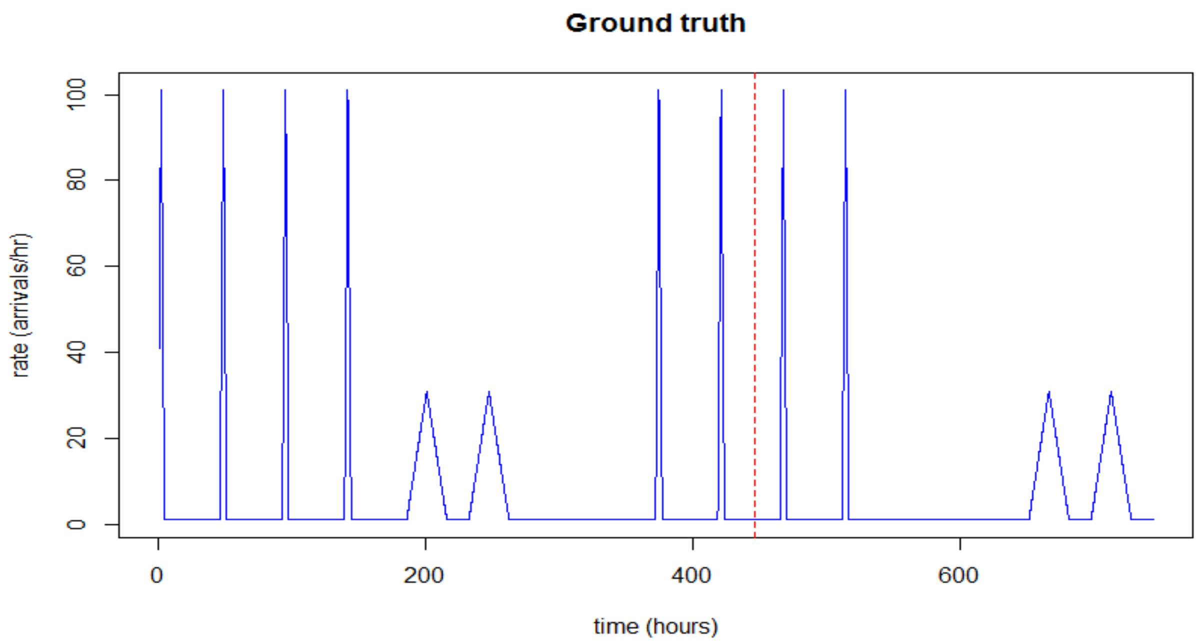
Existing models: Time Series (TS) Analysis

- Models temporal probability/rate of words
- Divide into fixed time buckets -> count the occurrences in each bucket -> model the most recent bucket against the previous ones

○ Drawbacks:

- fixed time intervals doesn't make much sense (next slide)
- it is not **time-scale invariant** (defn: timestamps N times denser ⇒ estimates N times larger)
- could give negative estimates

Why TS is not a good idea



- Imagine for TS:
  - if bucket too wide: cannot track changes responsively
  - if too narrow: many 0's in a row ⇒ hard to get a reasonable nonzero prediction
  - hard to choose a universal bucket size

Our idea & Assumption

- Data: a sequence of timestamps
  - TS: divide the sequence into batches of fixed time intervals <- seems unnatural
  - Our idea: analyze the original sequence of timestamps directly

--> Instead of using time series, use "sample series" (next slide)

--> Instead of updating every day/hr/min, update whenever we have new sample (new search)

- Assumption:

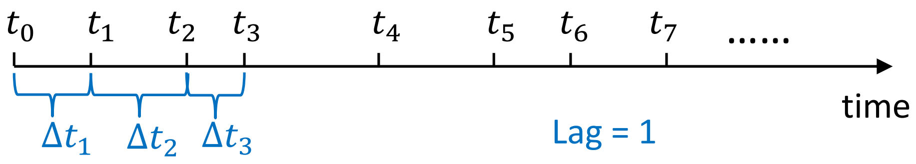
Inhomogeneous Poisson Process (a Poisson process with rate parameter $\lambda(t)$ a function of time).

- Goal: recover $\lambda(t)$

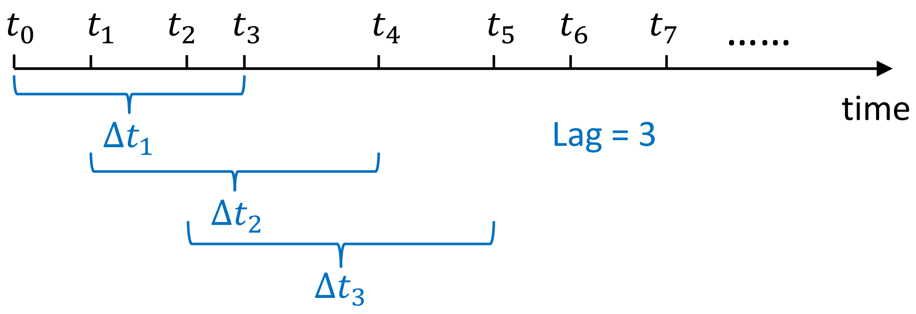
Sample series

- Given a seq of timestamps, can we use the temporal info directly? Sure.
- **Sample series**: instead of using fixed time buckets, use fixed counts buckets

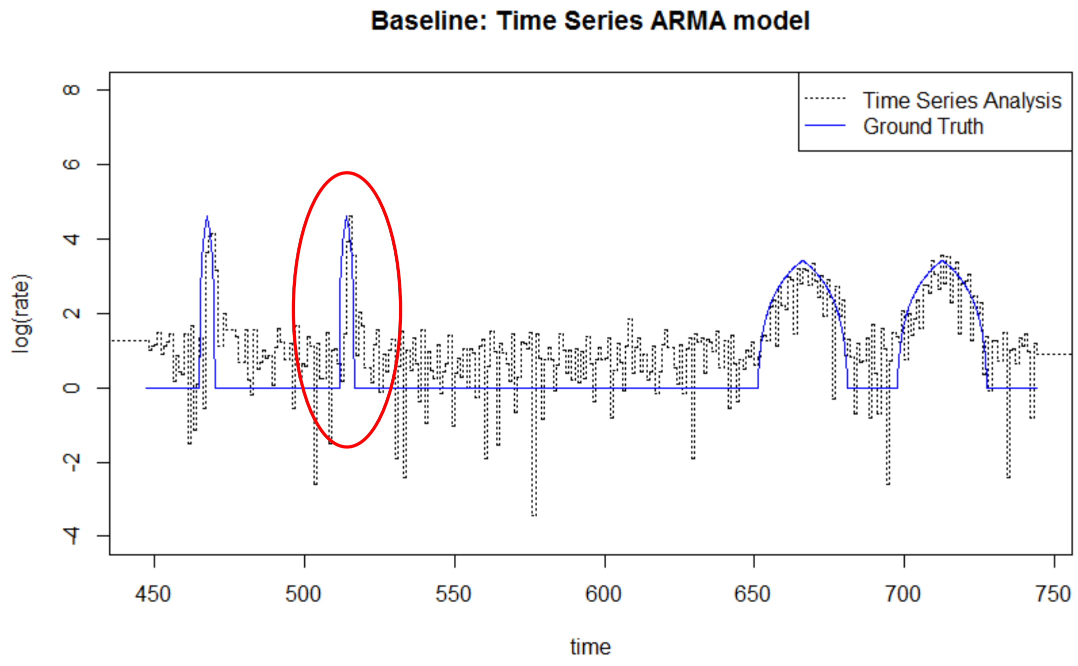
Sample series with lag 1:

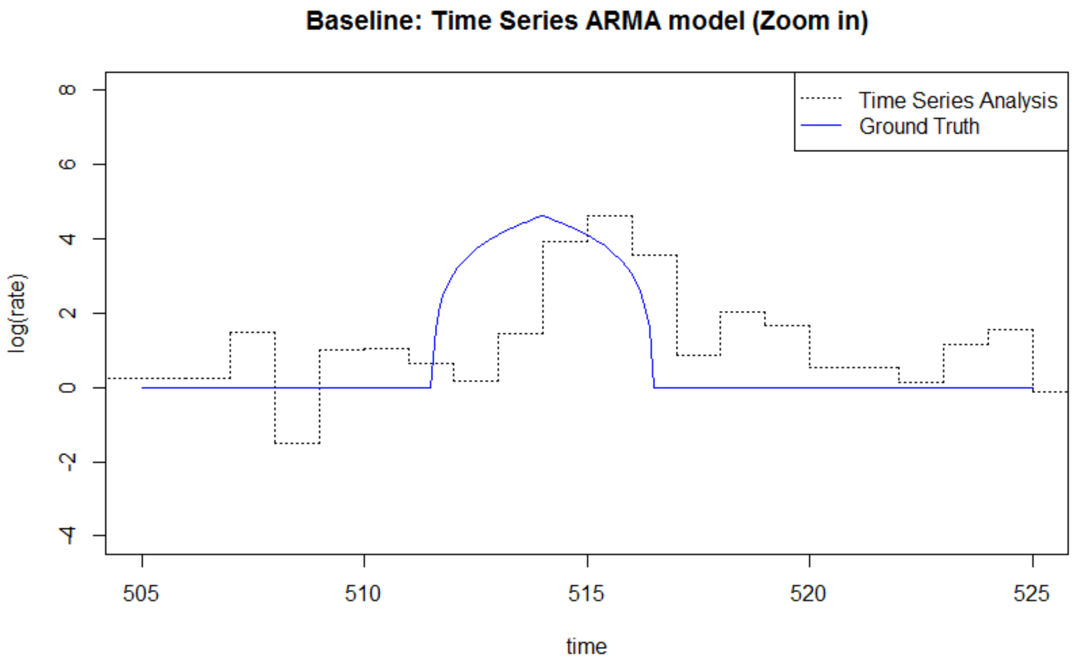


Sample series with lag 3:



Baseline: TS ARMA model



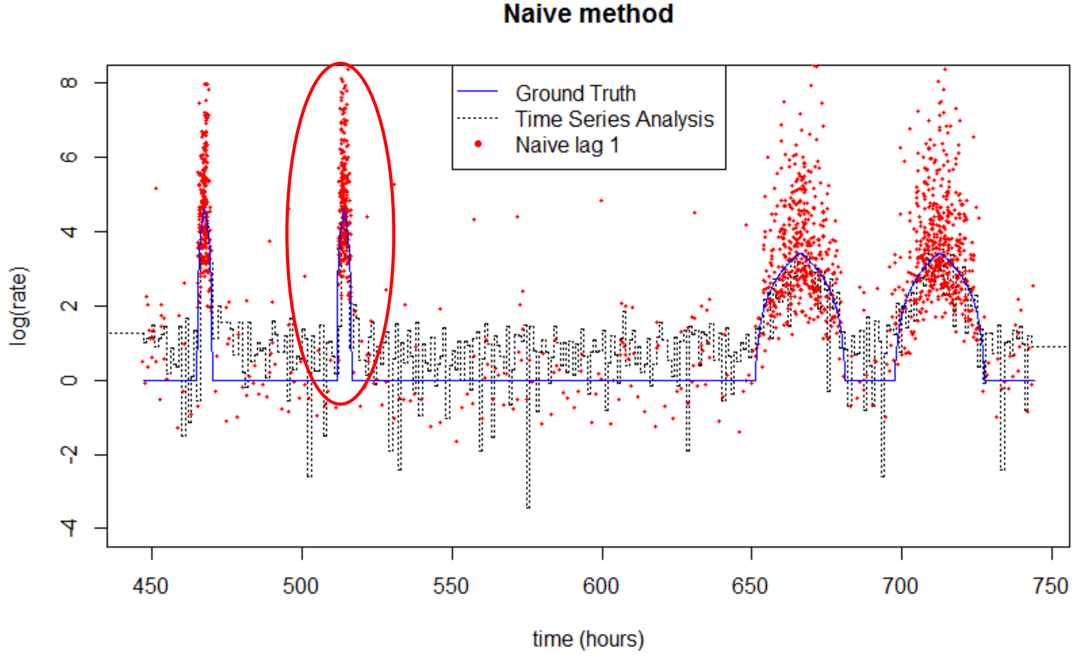


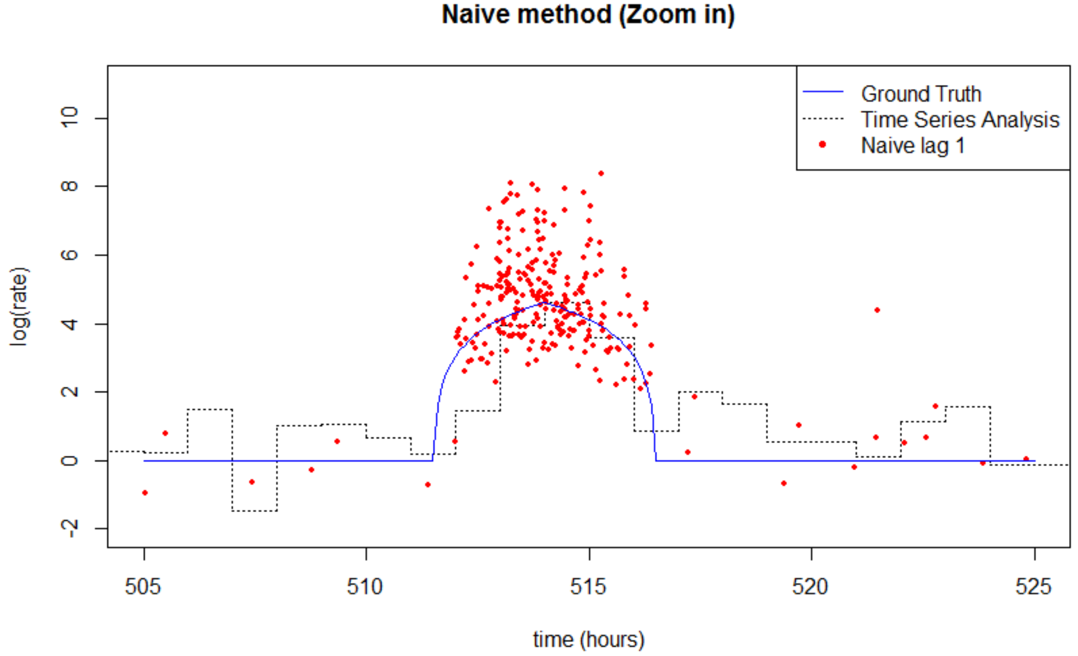
Cannot capture the change well!

Naive method

- Instantaneous rate est with lag = 1:

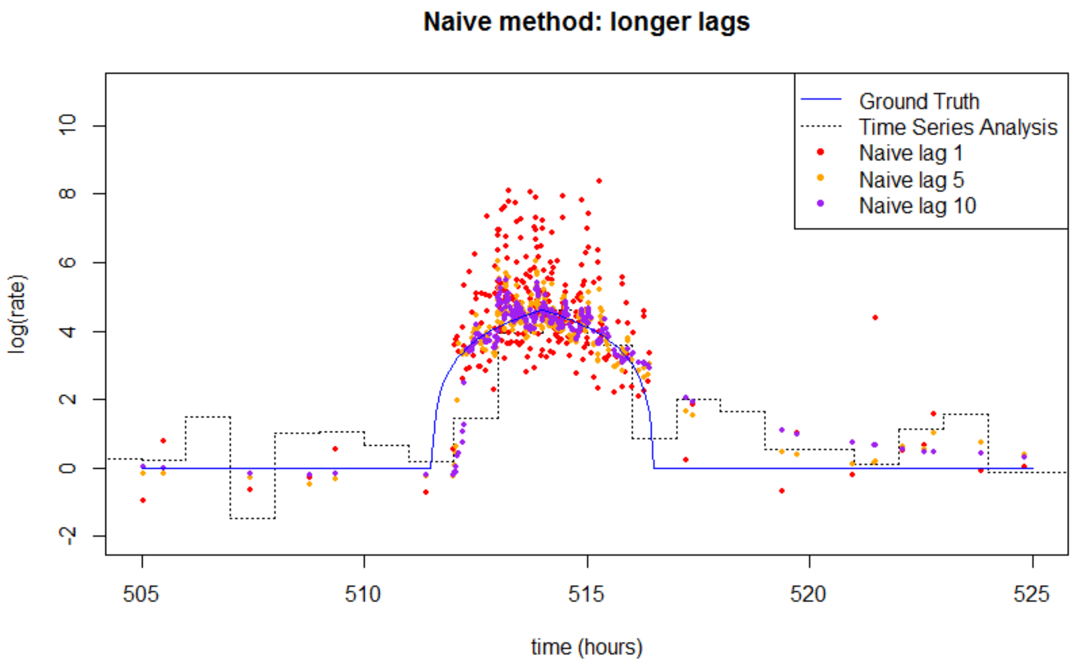
The est at the n-th timestamp is $\hat{\lambda}_n = 1/(t_n - t_{n-1})$





Responsive but very noisy! Try longer lags?

Naive method: longer lags

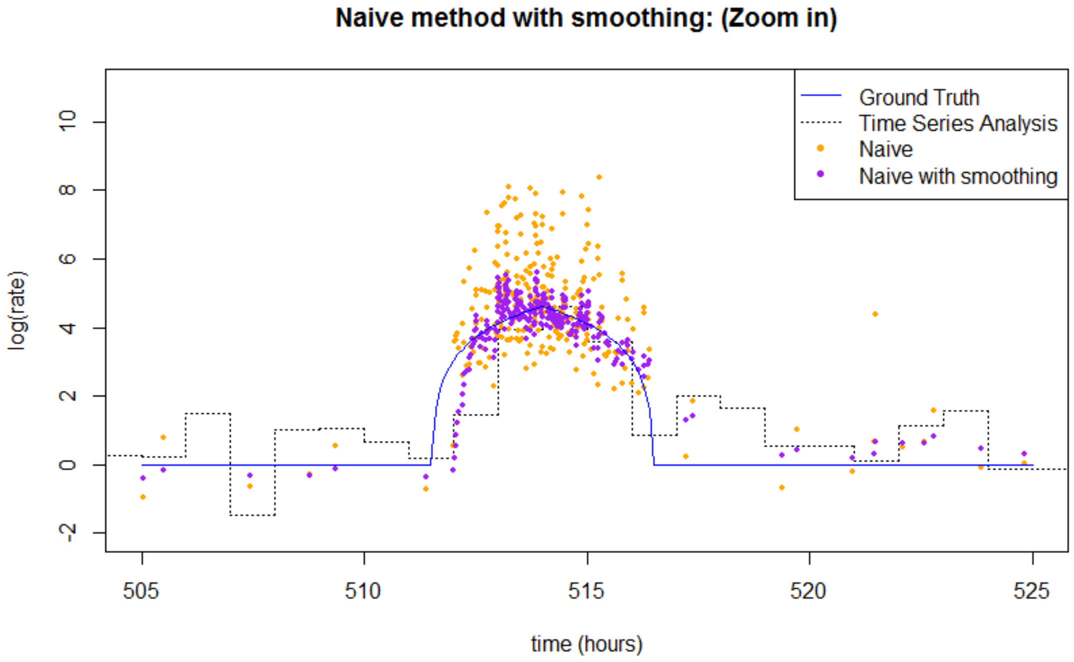


Naïve lag 5: $\hat{\lambda}_n = 5/(t_n - t_{n-5})$

Naïve lag 10: $\hat{\lambda}_n = 10/(t_n - t_{n-10})$

- lag too small ⇒ too **noisy**; lag too big ⇒ cannot capture sudden drop quickly
- Need some smoothing technique! But not the traditional smoothing technique since we are in the sample domain now.

Naïve method with smoothing

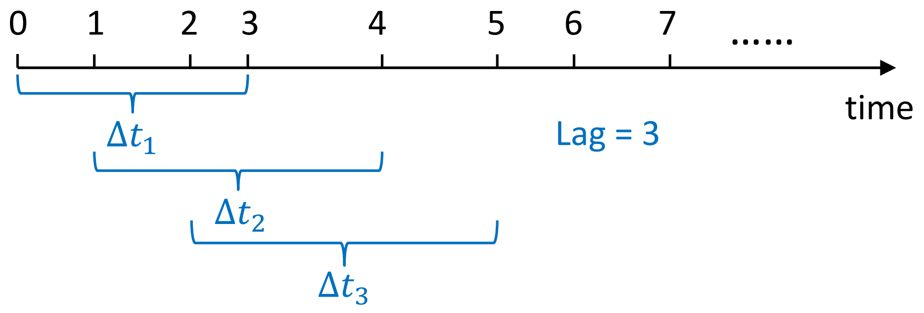


$$\hat{\lambda}_n = \frac{\sum_{i=0}^{L}(t_{n-i} - t_{n-i-1})\lambda_{n-i}w(n,i)}{\sum_{i=0}^{L}(t_{n-i} - t_{n-i-1})w(n,i)}$$

$$w(n,i) = e^{-i\theta}$$

Less noisy, but not more responsive

Bayesian method



Sequential Bayes with remembering factor ϕ:

- **Prior**: $\lambda \sim p^n(\lambda) \coloneqq p^{n-1}(\lambda|\Delta t_{n-1})$
- **Likelihood**: $\Delta t_n|\lambda \sim \Gamma(\mathrm{lag}, \lambda)$
- **Posterior**: $p^n(\lambda|\Delta t_n) \propto p(\Delta t_n|\lambda)[p^n(\lambda)]^\phi \propto \Gamma(\alpha_n,\ \beta_n) \Longrightarrow \hat{\lambda}_n = \frac{\alpha_n}{\beta_n}$

- It is scale invariant
- Bayes with lag 1, flat prior $\Longleftrightarrow$ naïve with smoothing

Bayesian method: continuous est.

- When nothing happens after a while $\Longrightarrow$ decrease our estimate
- Suppose:

(a) the posterior at t_n is $\Gamma(\alpha_n, \beta_n)$, $\hat{\lambda}(t_n) = \frac{\alpha_n}{\beta_n}$

(b) the word has not be searched during $(t_n, t_n + t]$,

then we update posterior at time $t_n + t$ as:

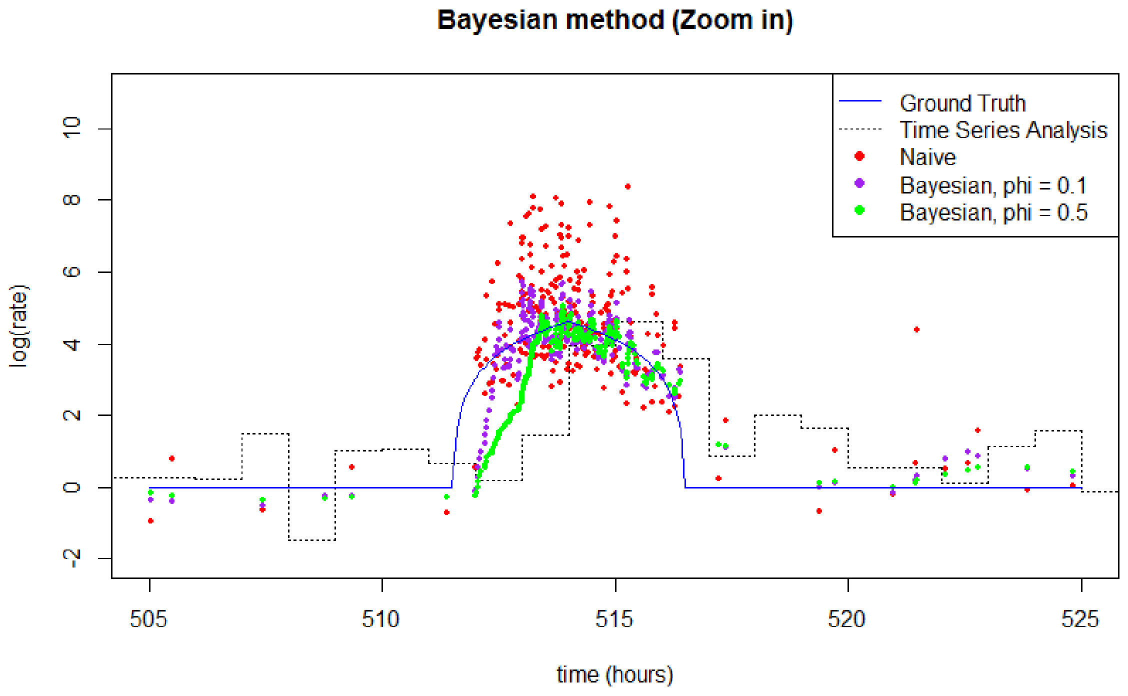
$$\widetilde{p^n}(\lambda|\Delta t_n) \propto e^{-\lambda t}\, p^n(\lambda|\Delta t_n) \propto \Gamma(\alpha_n, \beta_n\text{+t})$$

$\Longrightarrow$ posterior mean $\hat{\lambda}(t_n + t) = \frac{\alpha_n}{\beta_n + t}$

$\Longrightarrow$ we can estimate $\lambda(t)$ for any t

Bayesian method

- How does the choice of ϕ(remembering factor) affect the smoothing result?



Larger ϕ smoother, but less responsive

Generic Method

- Bayes is better than TS. Can we do better?
- A more generic form, optimize w.r.t. its para.
- Criterion: the **likelihood** of generating the entire seq of timestamps given the estimated $\hat{\lambda}(t)$.

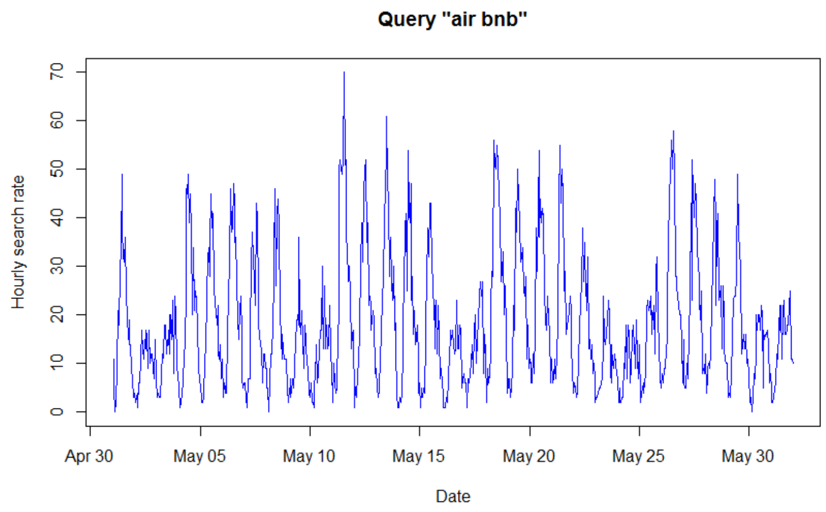
- Proposed generic form:

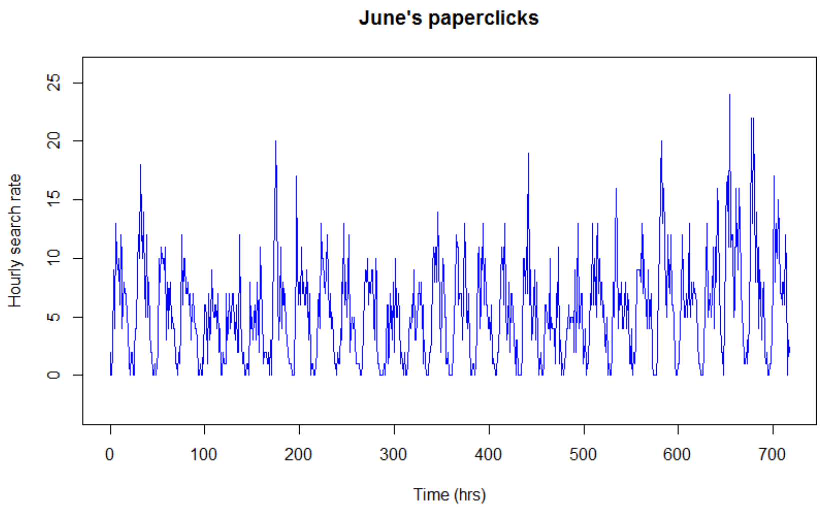
$$\lambda(t_n) = \frac{\alpha_n}{\beta_n}$$

$$\lambda(t_n + \Delta t) = \left(1 - e^{-\lambda \Delta t}\right)\frac{\alpha_n}{\beta_n + \Delta t} + e^{-\lambda \Delta t}\frac{\alpha_n}{\beta_n}$$

Real Data results

- ❑ Bing search query:
  - 2015-05-01 ~ 2015-05-31 queries with at least 7692 entries, with 1/64 random sampling
- ❑ Paper Click Data:
  - 2015-04-01 ~ 2015-06-30 (URLs associated with) papers with at least 400 clicks.
- ❖ Use 2/3 data for training and 1/3 data for testing





Real Data results: (Preliminary) Bing search query/Paper clicks

	TS			Bayesian			Generic		
Pdf score	optim	1 min	1 hr	$\phi = \phi_o - 0.1$	$\phi = \phi_o$	$\phi = \phi_o + 0.1$	$\phi = \phi_o - 0.1$	$\phi = \phi_o$	$\phi = \phi_o + 0.1$
Bing search	1.534	0.561	1.456	1.638	**1.647**	1.596	1.637	**1.648**	1.601
Paper clicks	-7.267	-22.53	-12.85	-1.005	**-1.008**	-1.027	-1.005	**-1.008**	-1.026

Summary: Part I

- Traditional way of thinking (fixed time buckets) doesn’t quite make sense
- “sample series”
- Naïve method: responsive but noisy
- Bayesian method: less noisy than naive, but can do better
- Generic method: less noisy than Bayesian

Part II:
Dynamic Language Modeling

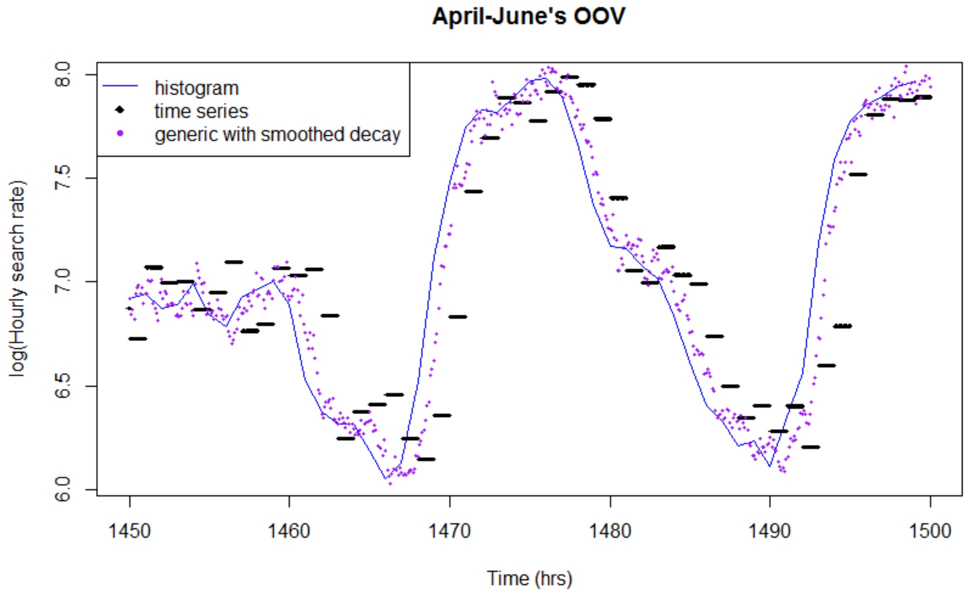
OOV Analysis

- OOV: items not seen in the training data (will be given a prob. of 0 without smoothing)
- a very important yet not perfectly addressed issue in LM
- existing methods: “add-one” smoothing, assign prob. mass to OOV(unseen items), …
- occurrences of OOV: a sequence of timestamps -> can also be modeled using “sample series” technique

- ❑ Real Data Analysis: Paper Click Data

- 2015-04-01 ~ 2015-06-30 clicks on URLs associated with papers
- Use 2 months data for training and 1 months data for testing

OOV Analysis: Results



A good thing: $\phi_{opt} \approx 0.995$ very consistently across all data

	TS			Our method		
Different scenarios	In ½ min	In min	In hr	$\phi = 0.9$	$\phi = 0.995$	$\phi = 0.999$
likelihood score	6.242	6.243	6.190	6.227	**6.250**	6.238

Our proposed method of doing Dynamic Language Modeling

- Dynamic Language Modeling if follow the traditional ideas:

➢ Example: $\underbrace{\text{AAABAACBDCC}}_{\text{training}}$ | $\underbrace{\text{AEBE}}_{\text{testing}}$: (suppose use "add-one" smoothing for oov)

The prob. for the specific testing seq is: $\frac{5}{12} \cdot \frac{1}{13} \cdot \frac{2}{14} \cdot \frac{2}{15}$

- Drawbacks:

  - does not use temporal info
  - different orderings lead to same est.
  - does not deal with OOV well

Our proposed method of doing Dynamic Language Modeling

- If $paper_i \sim Exp(\lambda_i), i = 1, \dots, M$, then

$$P(paper_i \ is\ clicked\ |a\ click\ happens) = \frac{\lambda_i}{\lambda_1 + \cdots + \lambda_M}$$

- Proposed metric (i.e. prob. of generating the whole sequence) is:

$$P\left(observing\ clicks\ on\ paper\ i_{t_1}, \dots, i_{t_N}\ at\ time\ t_1, \dots, t_N\right) =$$

$$= \prod_{n=1}^{N} \left(I[i_{t_n}\ is\ not\ OOV] \cdot \frac{\lambda_{i_{t_n}}(t_n)}{\sum_{i=1}^{M} \lambda_i(t_n)} + \mathrm{I}[i_{t_n}\ is\ OOV] \cdot prob_{OOV}\right)$$

✓ We are incorporating time information (as opposed to treating them as a bag of words)

✓ We are modeling rates

✓ We are incorporating OOV analysis

Comparison of our method and traditional LM

- Paper Click Data:
  - 2015-04-01 ~ 2015-06-30 clicks on URLs associated with papers
  - Training: April + May; Testing: June (first week of June)

	Traditional Dynamic LM		Our method	
Training Set	May	April + May	May	April + May
Prob. score	-13.74	-14.18	**-12.54**	-13.41

Summary

- Temporal info is valuable
- Bayesian and more generic method that fully used temporal info have better performances
- Applications in Dynamic Language Modeling; OOV Analysis

Acknowledgements

- Paul Hsu
- Yang Song
- Everyone in ISRC

Thank you!