Web science

Publication Date Prediction through Reverse Engineering of the Web

Matsuev Egor

2019

Overview

Motivation:
publication dates are often used as features in web search ranking

Problem:
detection of a web page publication date as not all web pages contain the publication dates in their texts and it is hard to distinguish the publication date among all the dates found in the page's text.
Solution:
novel link-based methods, some of them are based on a probabilistic model of the Web graph structure evolution

Methods

Extraction of publication date

- this method can be applied only to pages which contain the publication dates in their texts or URLs.

Link-based methods

- connected web resources tend to have similar update patterns so one could analyse publication dates of their neighbors
- backlinks are used to estimate the publication date of URI (shortening the URI, tweeting, public appearence)

Language models

- error rates are often measured in years, decades or even centuries

Anchor and seed dates extraction

The following locations are considered in order:

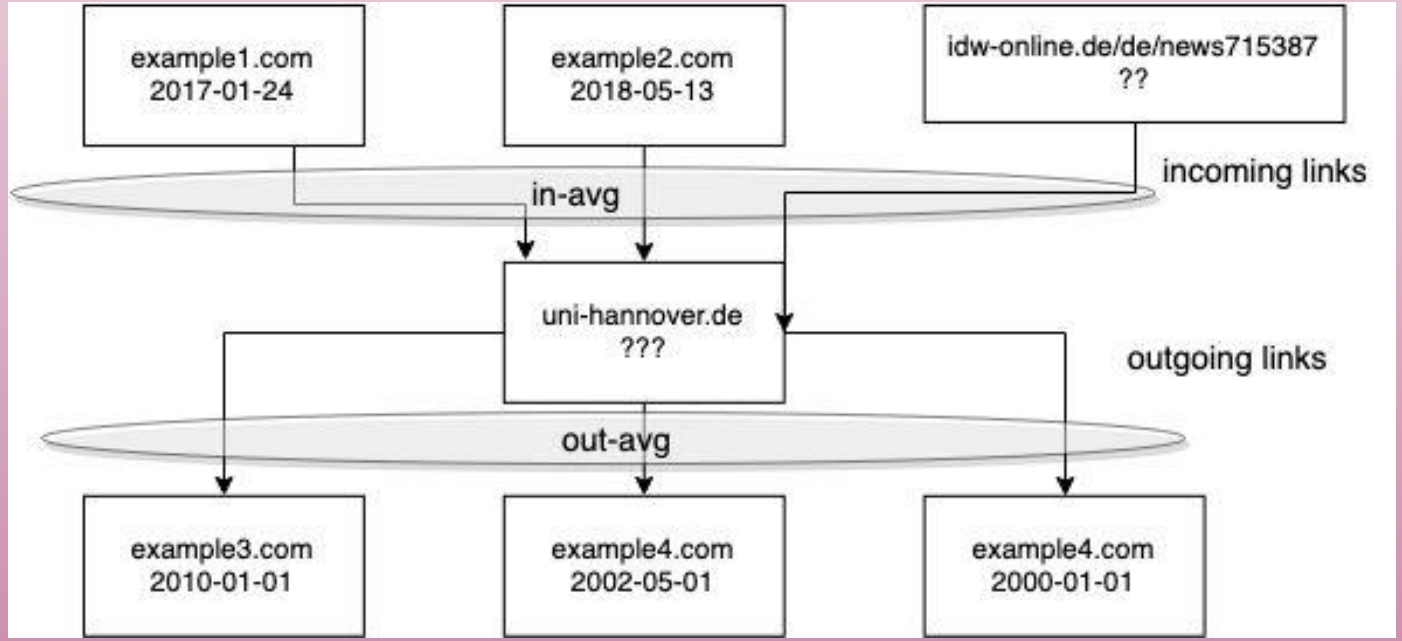
1. URL
2. Title
3. Just before main content
4. Just after main content
5. Main content of a document
6. All other locations

Location	mean / median / weighted err.	coverage (by non-constant)
Constant date	124 / 113 / 159	0%
+ URL	95.5 / 94.2 / 145	19.6%
+ before main content	87.3 / 92.1 / 103	20.1%
+ title	80.0 / 76.3 / 95.7	20.6%
+ after main content	79.1 / 75.4 / 95.2	20.8%
+ main content	78.8 / 74.0 / 87.7	21.3%
+ other	59.1 / 35.2 / 72.7	47.2%

Table 1: Errors for different seed dates on D_1

Date propagation

- **in-avg**
- **out-avg**
- **in-min**
- **all-avg**
- **out-max**
- **out-max-in-min**
- **model-q**



Model description

we mostly work with intra-site links, therefore here we describe the model with only one host
age difference a_ p,r for the pages p and r, that is, t_p − t_r
the page r is chosen from the page p, q_r- intrinsic quality of the page
lambda- attractiveness decay

$$\mathrm{attr}(q_r, a_{p,r}) = \begin{cases} q_r \cdot e^{-\lambda a_{p,r}} \cdot \left(1 - \frac{e^{-c a_{p,r}}}{2}\right) & \text{if } a_{p,r} \geq 0\,, \\ q_r \cdot e^{-\lambda a_{p,r}} \cdot \frac{e^{c a_{p,r}}}{2} & \text{if } a_{p,r} < 0\,. \end{cases}$$

Model description

Problem of the previous model edges going from older pages to newer ones are prohibited(for example page was updated)
A new model is proposed

$$f(x) = \begin{cases} 1 - \frac{e^{-cx}}{2} & \text{for } x \geq 0\,, \\ \frac{e^{cx}}{2} & \text{for } x < 0\,. \end{cases}$$

Old model:

$$\mathrm{attr}(q_r, a_{p,r}) = \begin{cases} q_r \cdot e^{-\lambda a_{p,r}} \cdot \left(1 - \frac{e^{-c a_{p,r}}}{2}\right) & \text{if } a_{p,r} \geq 0\,, \\ q_r \cdot e^{-\lambda a_{p,r}} \cdot \frac{e^{c a_{p,r}}}{2} & \text{if } a_{p,r} < 0\,. \end{cases}$$

Likelihood optimization

$$W(t_i) = \sum_{j \le i} q_j e^{-\lambda(t_i - t_j)} \left(1 - \frac{e^{-c(t_i - t_j)}}{2}\right) + \sum_{j > i} q_j e^{-\lambda(t_i - t_j)} \frac{e^{c(t_i - t_j)}}{2}.$$

$$L(\bar{t}, \bar{q}) = \prod_{ij \in G_{real}, i \ge j} \frac{q_j e^{-\lambda(t_i - t_j)} \left(1 - \frac{e^{-c(t_i - t_j)}}{2}\right)}{W(t_i)} \cdot \prod_{ij \in G_{real}, i < j} \frac{q_j e^{-\lambda(t_i - t_j)} \frac{e^{c(t_i - t_j)}}{2}}{W(t_i)},$$

1. fix lambda and c
2. maximize likelihood function
3. get publication time t

gradient descent method is used

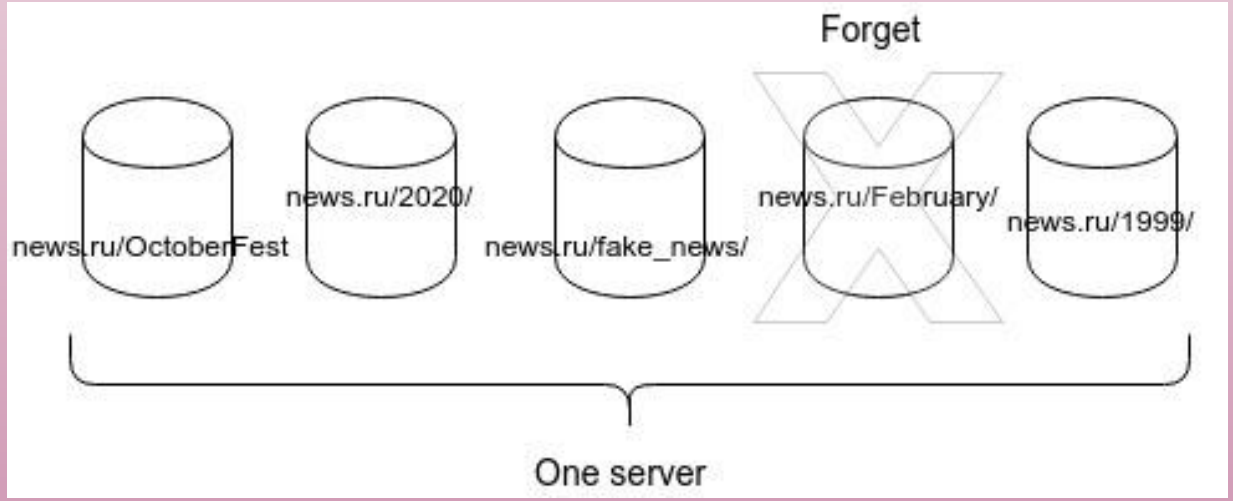
Simplification of likelihood optimization

It was shown that W (t_i) is asymptotically a constant. So, we can replace W (t_i) in the likelihood function L(ŧ, q̄) by a constant W

$$\prod_{pi \in G_{real}} \mathrm{P}(pi \in G_{model}) \cdot \prod_{jp \in G_{real}} \mathrm{P}(jp \in G_{model})$$

$$= \prod_{pi \in G_{real}} \frac{e^{-\lambda(t_p - t_i)} f(t_p - t_i)}{W} \cdot \prod_{jp \in G_{real}} \frac{e^{-\lambda(t_j - t_p)} f(t_j - t_p)}{W} \cdot$$

Data-driven parameter selection



- compare the estimated dates with the actual anchor dates and get mean absolute error
- choose the best combination of parameters
- find an optimal number of steps as stopping criteria

Data

- Crawled a by Yandex (search engine) from January 2013 to May 2014. 4M pages from 70 hosts
- Memetracker dataset 96M blog posts and news articles published during 9 months

Results

	Method	mean/median/ weighted err	coverage
baselines	seed and anchor dates	57.1/30.2/70.0	47.7%
	1-step in-avg	55.9/29.8/56.9	49.3%
	1-step out-avg	56.3/27.6/68.2	64.1%
	1-step in-min	58.0/29.8/66.0	49.3%
proposed methods	1-step all-avg	55.5/27.7/57.0	64.4%
	1-step out-max	56.9/28.1/67.5	64.1%
	1-step out-max-in-min	56.7/30.3/70.0	49.0%
	in-avg	55.5/29.7/56.5	49.7%
	out-avg	58.0/25.0/70.1	68.6%
	all-avg	52.4/22.3/55.6	69.7%
	in-min	57.8/29.7/65.8	49.7%
	out-max	57.1/22.3/72.7	68.6%
	out-max-in-min	56.6/30.0/70.2	49.4%
	model-0.5	51.9/19.7/53.9	69.7%
	model-0.6	**51.2**/19.7/**53.6**	69.7%
	model-0.7	51.4/**19.5**/53.7	69.7%
	likelihood optimization	**49.9/19.1/51.2**	69.7%

Table 2: Comparison of the algorithms on D_2

	Method	mean/median/ weighted err	coverage
baselines	anchor dates	77.6/79.7/89.6	50.0%
	1-step in-avg	71.0/73.3/39.7	57.9%
	1-step out-avg	66.8/66.6/41.7	65.5%
	1-step in-min	71.5/73.6/77.8	57.9%
proposed methods	1-step all-avg	64.1/62.3/38.6	68.6%
	1-step out-max	67.4/67.0/50.0	65.5%
	1-step out-max-in-min	73.5/75.8/58.0	54.8%
	in-avg	70.2/72.5/39.0	58.9%
	out-avg	64.5/62.4/39.6	69.0%
	in-min	70.9/73.0/77.2	58.9%
	all-avg	**58.0**/52.6/37.9	75.3%
	out-max	65.5/63.1/48.1	69.0%
	out-max-in-min	72.7/75.2/56.3	55.8%
	model-0.3	58.3/52.3/43.0	75.3%
	model-0.4	**58.0/52.0**/38.1	75.3%
	model-0.5	58.2/52.2/**35.7**	75.3%
	likelihood optimization	**57.4/51.3/33.4**	75.3%

Table 3: Comparison of the algorithms on MemeTracker

Results

Model-q propagation shows the best results according to all metrics
Likelihood optimization allowed us to get better results according to all measures.
Finally, we get 11%, 31%, and 10% improvements of mean, median, and weighted errors respectively over the best of the baselines.

Dataset size influence

a sparser link structure leads to worse performance of the algorithms
It turns out that the weighted error is much less sensitive to the sparsity in data than the mean absolute error.

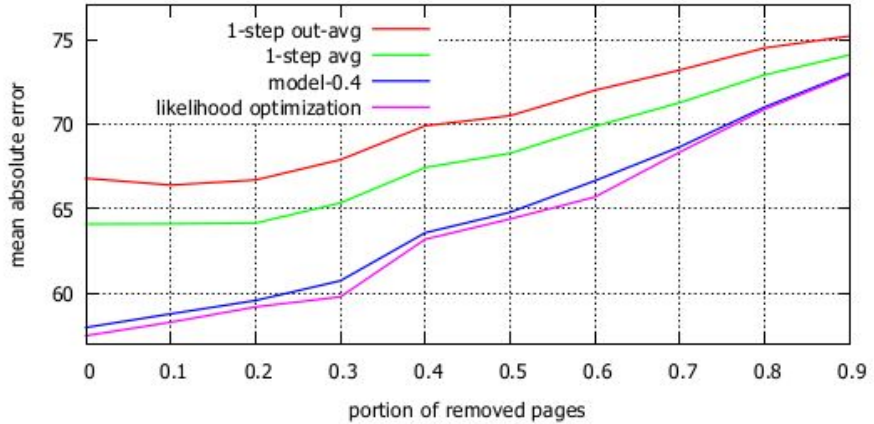


Figure 1: Influence of sparsity in data: mean absolute error

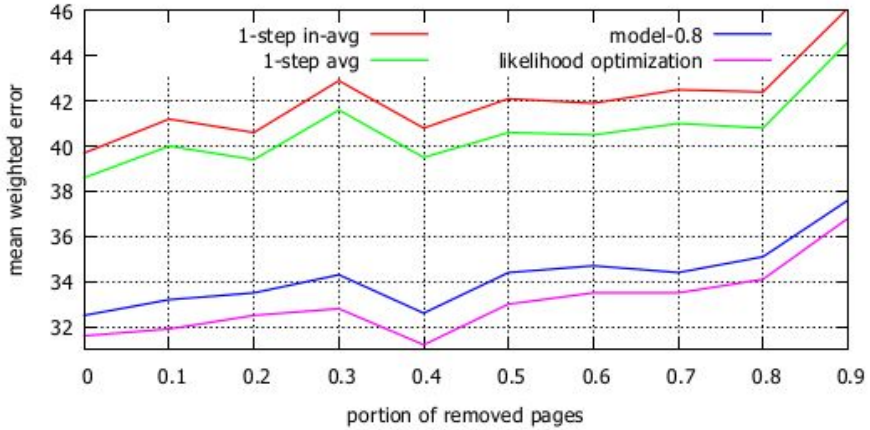


Figure 2: Influence of sparsity in data: mean weighted error

Thank you for your attention!