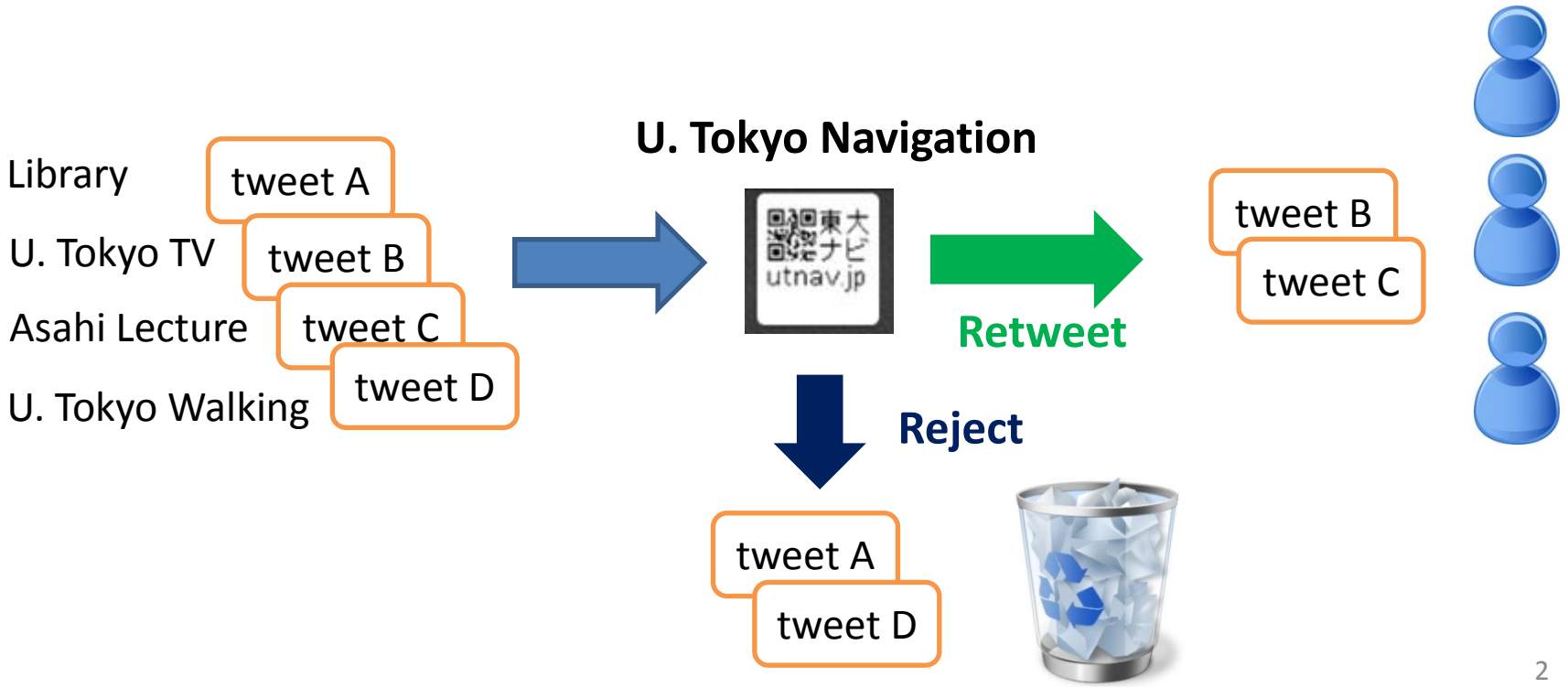
Online Retweet Recommendation with Item Count Limits

Xiaoqi Zhao, Keishi Tajima
Kyoto University

Background

- Portal accounts on Twitter
  - Instead of posting original tweets, these accounts retweet tweets which are useful to their followers.
  - Similar to the portal sites on the web.



Problem

Conditions for good portal accounts:

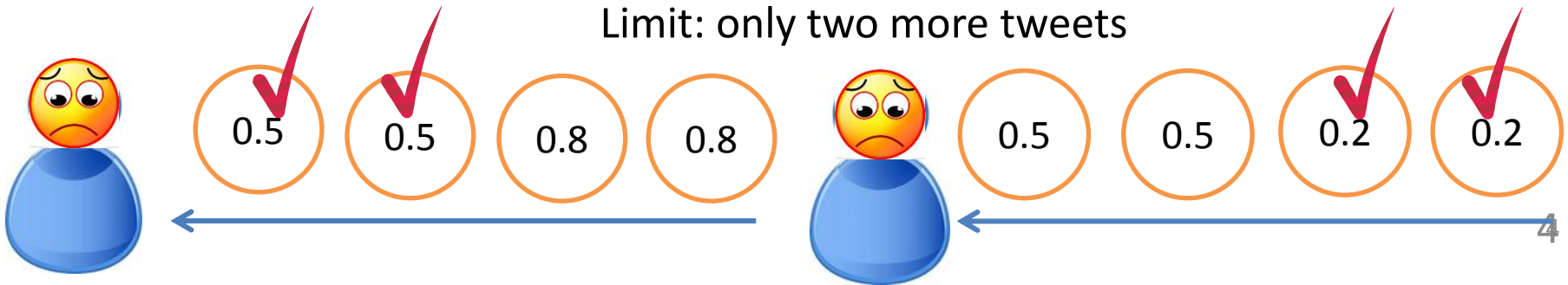
1. Choose tweets that meet followers' interest.
2. Retweet timely.
   - Life cycle of a tweet is only 48 hours [**Fritz**,2013]
3. Retweet an appropriate number of tweets.

A possible solution:

- Online Retweet Recommendation with Item Count Limits

Online Retweet Recommendation with Item Count Limits

- Recommend a given number of tweets.
- Recommended tweets cannot be canceled.
- Difficult Point: **Trade-off between quality and delay**
  - For better selection, we need to know the following tweets.
  - For timely recommendation, we have to make decisions before seeing them.



Previous Work

- Many studies on tweet recommendation
- A few studies on retweet recommendation

E.g.: User oriented tweet ranking: a filtering approach to microblogs [UYSAL, et al., CIKM 2011]

None of them considers the upper bound of the number of tweets to recommend.

Solutions In Our Paper

We propose and compare 4 algorithms:

- Real-time Recommendation (Online)
  1. History-Based Threshold Algorithm
  2. Stochastic Threshold Algorithm
- Non-real-time Recommendation (Semi-Online)
  3. Time-Interval Algorithm
  4. Every-k-Tweets Algorithm

Related Problem:
Multiple-choice Secretary Problem
[Robert Kleinberg et al. 2005]

Input:

- A sequence of secretaries $T = t_1, t_2, \cdots, t_n$
  - Size of the sequence: n
  - Score v: $T \rightarrow R^+$
  - C: Number of secretaries to select

Output:

- A sequence of selected secretaries $O \subseteq T$
  s.t. maximizes $v(O)$ subject to $|O| \leq C$

Online Retweet Recommendation with Item Count Limits

Input:

- A sequence of tweets $T = t_1, t_2, \cdots$
  - Size of the sequence: Unknown
  - Score v: $T \rightarrow [0,1)$
  - C: Item count limit

Output:

- A sequence of selected tweets $O \subseteq T$
  s.t. maximizes $v(O)$ subject to $|O| \leq C$

Algorithm Threshold [Kellerer et al. 2004]

Our Algorithms 1 & 2 are based on the following algorithm.

- **Initialization**
  - $O = an\ empty\ sequence$
  - Initialize Threshold θ
- **While a tweet t_i comes in the time period do**
  - If $|O| \leq C$ and $v(t_i) \geq \theta$ then
    - $O = O \cup \{t_i\}$
    - Update θ
- Return O

1. History-Based Threshold Algorithm

- We use the average of the best thresholds in pervious k time periods.

$k = 3$

The best thresholds in last 3 time periods

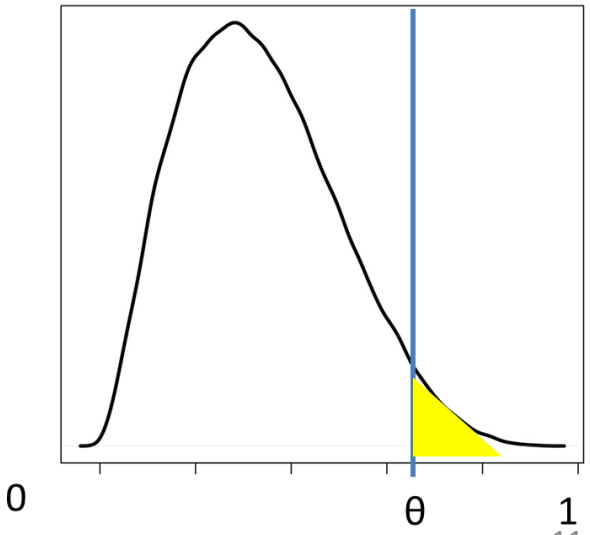
$$Threshold\ \theta = 0.7$$

2. Stochastic Threshold Algorithm

Two Assumptions are used to calculate the threshold:

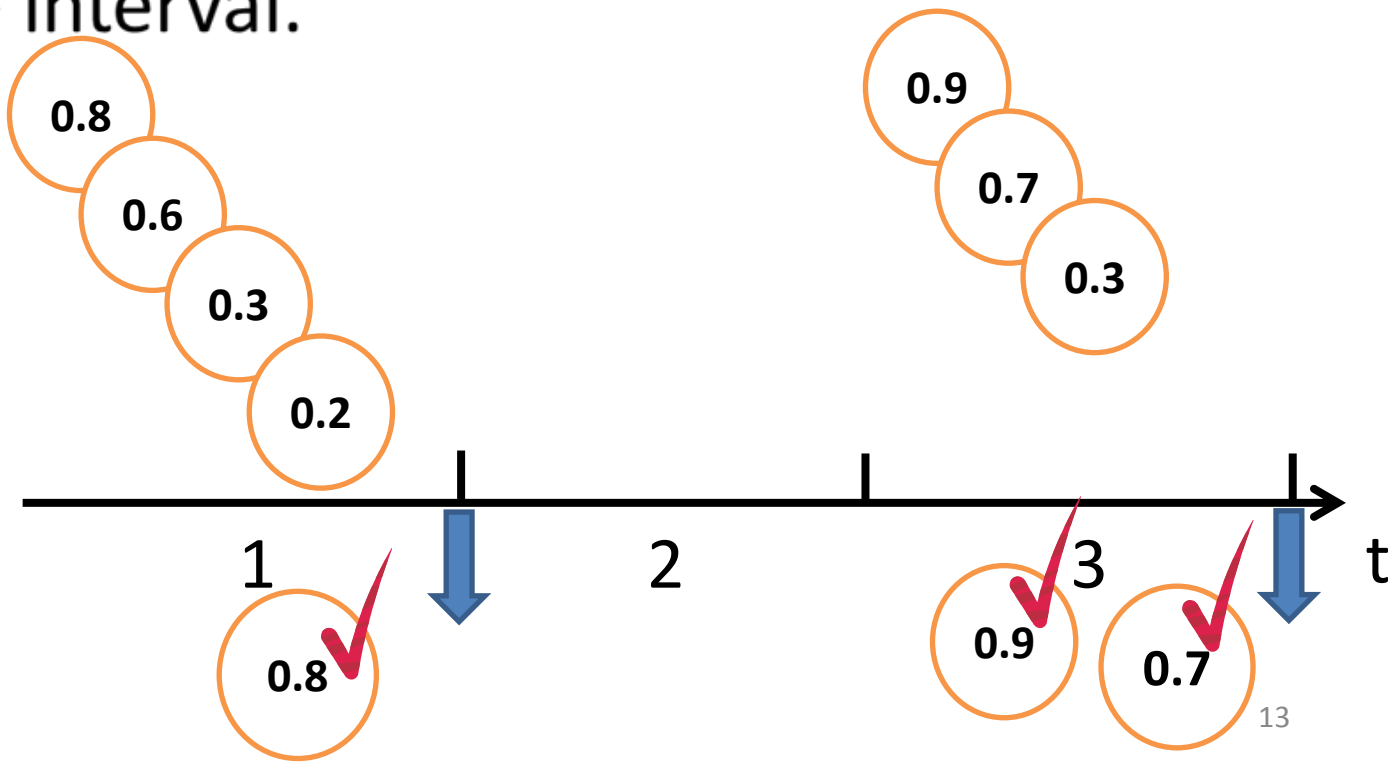
- Scores of incoming tweets$\sim B(x, a, b)$.
- Scores of tweets in a certain period are

random sampled from a given distribution.



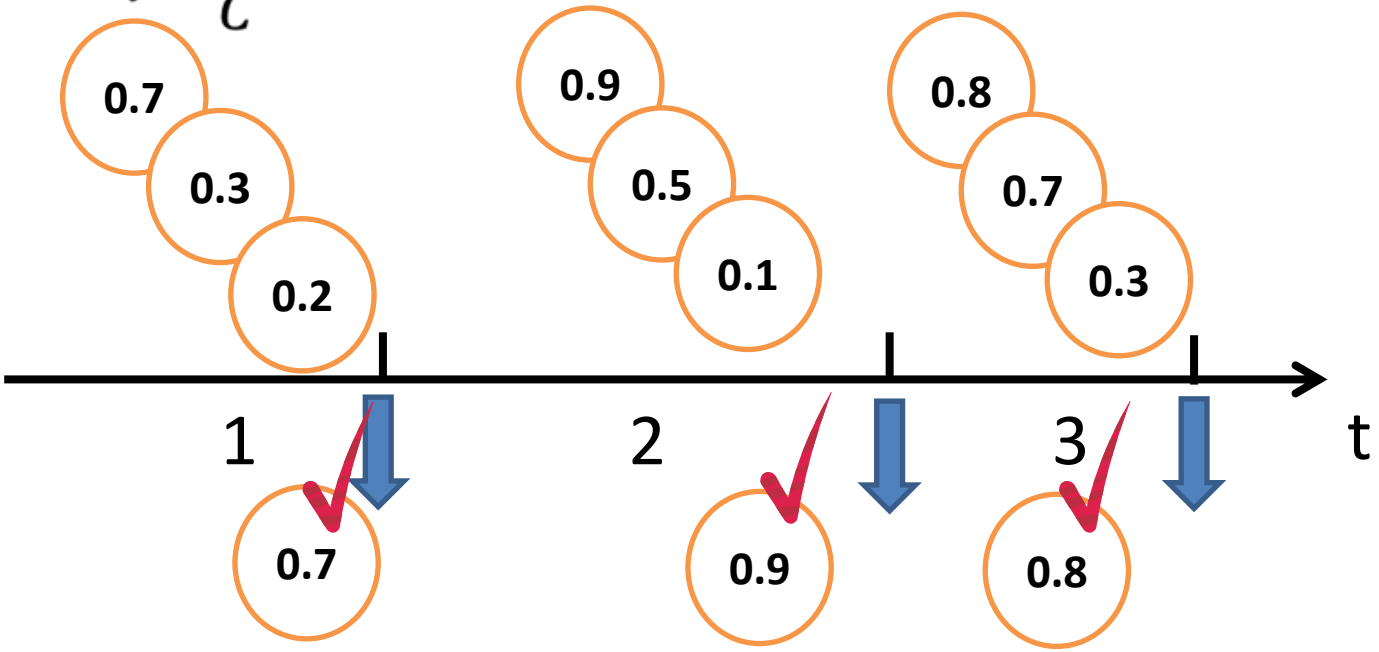
3. Time-Interval Algorithm

- Divide a given time period into C sub-intervals.
- Recommend top 1 tweet in each sub-interval.
- If nothing incomes, we carry it over to the next time interval.

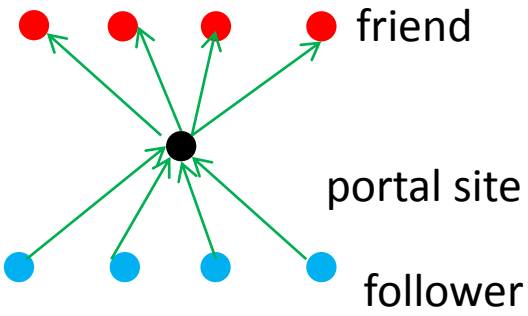


4. Every-k-Tweets Algorithm

- Use the average number $\widehat{|T|}$ of tweets in k pervious time period as the estimated number of incoming tweets.
- For every $\frac{\widehat{|T|}}{C}$ tweets, we select the best tweet.



Experiment



- Dataset:
  - 7 portal accounts
  - tweets from their friends and followers
- Item count limits in 24 hours: $C = 6,\ 12,\ 24$
- Evaluate 4 methods:
  1. History-Based Threshold Algorithm
  2. Stochastic Threshold Algorithm
  3. Time-Interval Algorithm
  4. Every-k-Tweets Algorithm

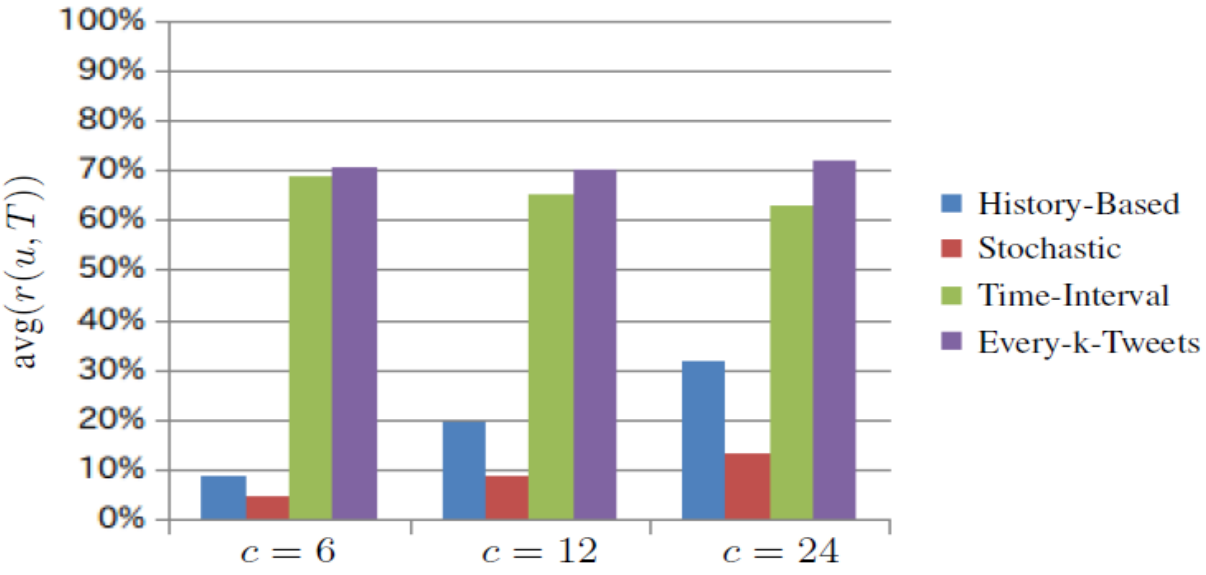
Experiment Evaluation

- Quality
  - Competitive ratio r :

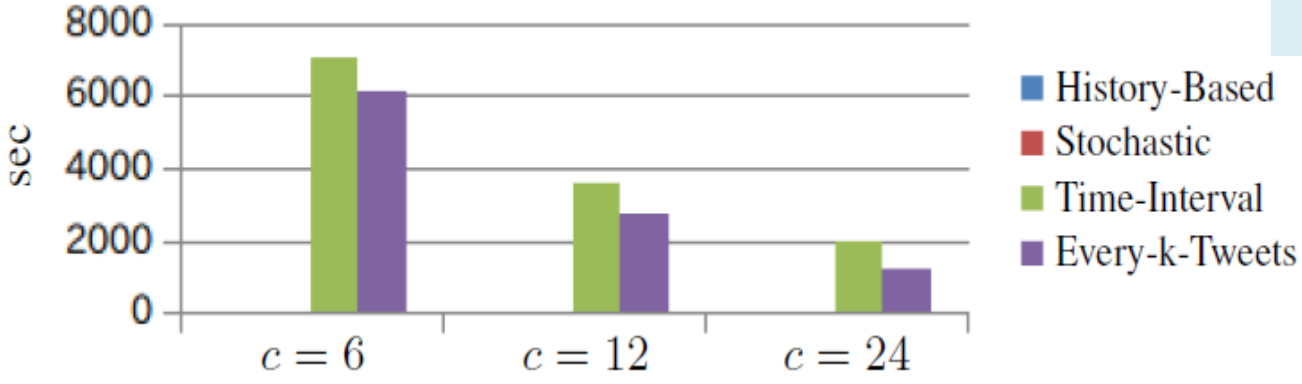
$$r = \frac{\sum_{t \in O_{online}} v(t)}{\sum_{t \in O_{offline}} v(t)}$$

- Delay
  - Time lag between receiving time and recommendation time

Experiment Result



Quality



Delay

Time-interval & Every-k-tweets

- Higher quality
- Acceptable delays

Experiment Result (Delay)

Delay (sec)	μ	σ	max	min
	$c = 6$			
Time-Interval	7051	64	14341	103
Every-k-Tweets	6084	90	52241	0
	$c = 12$			
Time-Interval	3609	45	7199	13
Every-k-Tweets	2716	69	44165	0
	$c = 24$			
Time-Interval	1946	33	3599	5
Every-k-Tweets	1198	48	32084	0

- Every-k tweets algorithm is better for the average delay.
- Time-interval algorithm is better for the maximum delay.

Conclusion

- We proposed four retweet recommendation algorithms for portal accounts.
- We tested them with real twitter data.
- The method proposed in the paper can be easily applied to item recommendation for other stream media and some information applications for smartphones.
  - RSS
  - Antenna