Meme Tracking

Abhilash Chowdhary
CS-6604
Dec. 1, 2015
Overview

- Introduction
  - Information Spread
  - Meme Tracking

- Part 1: Rise and Fall Patterns of Information Diffusion: Model and Implications

- Part 2: NIFTY: A System for Large Scale Information Flow Tracking and Clustering
Introduction

Information Spread/Flow
Introduction

Information Spread/Flow

• How does the information spread through time?
Introduction

Information Spread/Flow

• How does the information spread through time?
• How does it mutate as it spread?
Introduction

Information Spread/Flow

• How does the information spread through time?
• How does it mutate as it spread?
• How do we predict the rise and fall
Introduction

Information Spread/Flow

- How does the information spread through time?
- How does it mutate as it spreads?
- How do we predict the rise and fall?
- Does it follow SI Model?
Introduction
Meme Tracking

• Memes: Short textual phrases that travel and mutate through the Web.
• They can be #hashtags, phrases commonly used in news articles etc.

"you can put lipstick on a pig" (# of mentions in blogs)

"yes we can"

(per hour, 1 week)
Overview

• Introduction
  • Information Spread
  • Meme Tracking

• Part 1: Rise and Fall Patterns of Information Diffusion: Model and Implications

• Part 2: NIFTY: A System for Large Scale Information Flow Tracking and Clustering
Rise and Fall Patterns of Information Diffusion: Model and Implications

Outline
- Problem definition
- Proposed method
- Experiments
- Forecasting
- Conclusions
Problem Definition

Informal Definition

• Model/predict an activity (e.g., number of blog postings), as a function of time, given some breaking-news at a given timetick.
Problem Definition

Problem 1 (What - If?)

Given :

• Network of bloggers/users ($S_b = n_b$)
• External shock/event
• Quality of the event $\beta$

Find :

• How blogging activity will evolve over time
Problem Definition

Problem 2 (Model Design)

Given:
• Behavior of Spikes

Find:
• Equation/model that can explain them, e.g.,
  • # of potential bloggers - $S_b$
  • Strength of external shock - $S()$
  • Quality of the event $\beta$
Rise and Fall Patterns of Information Diffusion: Model and Implications

Outline

- Problem definition
- **Proposed method**
- Experiments
- Forecasting
- Conclusions
Proposed Method

SpikeM captures 3 behaviors of the spikes:

- **Power-Law fall pattern**
  - Previous methods fit to exponential fall contrary to real data (Long Tail)
- **Periodicity**
  - Bloggers may modulate their activity following a daily cycle (or weekly, or yearly)
- **Avoidance of infinity divergence**
  - For divergence, the population is forced to be finite
Proposed Method

SpikeM captures 3 behaviors of the spikes:

1. periodicities
2. avoid infinity
3. power-law fall
Main Idea

Nodes (bloggers) are of two types:
• $U$ = Uninformed of rumor/event
• $B$ = informed, and Blogged about rumor

Assumption:
• If a blogger is informed about event, he will blog about it.
Main Idea

State Transition with time:

- Time $n=0$; **Un-informed** bloggers
Main Idea

State Transition with time:
- Time $n=0$; **Un-informed** bloggers
- Time $n=n_b$; **External shock** at time $n_b$; $S_b$ bloggers are informed, blog about news
Main Idea

State Transition with time:
• Time $n=0$; **Un-informed** bloggers
• Time $n=n_b$; **External shock** at time $n_b$; $S_b$ bloggers are informed, blog about news
• Time $n=n_b+1$; **Infection** (word-of-mouth effects)
Main Idea

Infectiveness of a blog-post:

- $\beta =$ Strength of infection (quality of news)
- $f(n) =$ Decay function (how infective a blog posting is)

![Decay function graph](image)
Main Idea: SpikeM-base

Equations of SpikeM (base)

\[ \Delta B(n+1) = U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot f(n+1-t) + \varepsilon \]

\[ U(n+1) = U(n) - \Delta B(n+1) \]

N - Total population of available bloggers
β - Strength of infection/news
n_b, S_b - External shock at birth (time)
ε - Background noise

\[ S(n) = \begin{cases} 0 & (n \neq n_b) \\ S_b & (n = n_b) \end{cases} \]

\[ B(n) = \sum_{t=0}^{n} \Delta B(t) \]
Main Idea: SpikeM-with periodicity

Full equation of SpikeM

$$\Delta B(n+1) = p(n+1) \cdot U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot f(n+1-t) + \epsilon$$

$$U(n+1) = U(n) - \Delta B(n+1)$$

Blogged
Periodicity

Un-informed

Bloggers change their activity over time (e.g., daily, weekly, yearly)

12pm Peak activity
3am Low activity

$$p(n)$$

Time n
Main Idea : Model fitting

SpikeM consists of 7 parameters:

\[ \theta = \{N, \beta, n_b, S_b, \epsilon, P_a, P_s\} \]

Learning Parameters
- Given a real time sequence
  \[ X = \{X(1), \ldots X(n), \ldots, X(n_d)\} \]
- Minimize the error
  - (Levenberg-Marquardt (LM) fitting)

\[ D(X, \theta) = \sum_{n=1}^{n_d} (X(n) - \Delta B(n))^2 \]
Main Idea : Analysis

SpikeM matches reality
- exponential rise and power-law fall

*SpikeM vs. SI model* (susceptible infected model)
Rise and Fall Patterns of Information Diffusion: Model and Implications

Outline
• Problem definition
• Proposed method
• Experiments
• Forecasting
• Conclusions
Experiments

Results of SpikeM fitting on three hashtags from Twitter dataset.
Experiments

Results of SpikeM fitting on six patterns from MemeTracker dataset.
Rise and Fall Patterns of Information Diffusion: Model and Implications

Outline
• Problem definition
• Proposed method
• Experiments
• Forecasting
• Conclusions
Forecasting

“What-If” Forecasting
• SpikeM forecasts not only tail-part, but also rise-part!

![Diagram showing forecasting for movie releases](image)

e.g., given

1. first spike,
2. release date of two sequel movies
3. access volume before the release date
Forecasting

“What-If” Forecasting
• SpikeM forecasts not only tail-part, but also rise-part!

SpikeM can forecast upcoming spikes
Rise and Fall Patterns of Information Diffusion: Model and Implications

Outline
• Problem definition
• Proposed method
• Experiments
• Forecasting
• Conclusions
Conclusion

SpikeM has following advantages:

• **Unification power**
  It includes earlier patterns/models

• **Practicality**
  It Matches real datasets

• **Parsimony**
  It requires only 7 parameters

• **Usefulness**
  What-if scenarios, outliers, etc.
Overview

• Introduction
  • Information Spread
  • Meme Tracking

• Part 1 : Rise and Fall Patterns of Information Diffusion: Model and Implications

• Part 2 : NIFTY: A System for Large Scale Information Flow Tracking and Clustering
NIFTY: A System for Large Scale Information Flow Tracking and Clustering

Outline
• Introduction
• Goal
• Proposed method
• Experiments
• Conclusions
Introduction

• The real-time information changes dynamically and spreads rapidly through the Web.
• For building a system to handle it, we need to know the following
  - How information varies over time
  - How it is transmitted
  - How it mutates as it spreads
NIFTY: A System for Large Scale Information Flow Tracking and Clustering

Outline
• Introduction
• Goal
• Proposed method
• Experiments
• Conclusions
Goal

To develop a system which:

tracks information as it spreads & mutates over time periods spanning many years
NIFTY: A System for Large Scale Information Flow Tracking and Clustering

Outline
• Introduction
• Goal
• Proposed method
• Experiments
• Conclusions
Proposed Method

• News Information Flow Tracking, Yay! (NIFTY)
  - System for large scale real-time tracking of “memes”
  - Highly scalable meme-clustering algorithm
  - Identifies mutational variants of a single meme
Proposed Method

NIFTY overview : Meme Definition

- Memes are short quoted phrases in a given document (web page in this case)
- This is Intuitive:
  - Quotes are an integral of journalistic practice
  - Quotes might be found even in unrelated news story
  - And tend to travel as story evolves
- These are elements recognizable to consumers of the media
Proposed Method

NIFTY overview : Meme Mutation

- Memes tend to mutate over a period of time
- Even though the text may be different the essence is same
Proposed Method

NIFTY Pipeline

Figure 3: Overview of the NIFTY pipeline.
Proposed Method

NIFTY: Phrase/Meme/Quote Extraction

- Input to NIFTY is documents $D \in \{\text{Items from Web like news report or blog post}\}$
- Phrases/Quotes are extracted from each element in $D$
Proposed Method

NIFTY: Document and Phrase Filtering

- Input documents D have too much spam, duplicates and irrelevant information
- Two pass Filtering process
- First Pass Filter: Filtering documents and Phrases
  - Removes blacklisted urls in D, phrases with inappropriate length ($|l|<3$ or $|l|>50$) and phrases having lack of ASCII chars (50%)
- Second Pass Filter: Advanced heuristics
  - Infrequent phrases removed
  - Language filtering (English letter percentage)
Proposed Method

NIFTY: Phrase Clustering

Phrase Graph Creation : Phrase Distance

\[ P = \text{Phrase base} ; \ D = \text{Document base} \]

- Find similarity between pairs of phrases in \( P \)
- Phrase Distance : Substring Edit Distance
  - minimum number of word insertions, deletions or substitutions needed to transform one string into a substring of the other string.
Proposed Method

NIFTY: Phrase Clustering

Phrase Graph Creation : Edge Creation

Given a pair of phrases and their substring edit distance, determine whether one of them is derived from the other one and thus should be connected by an edge

- Train a decision tree over hand labeled pairs of phrases (mutually variant and invariant both!)
- If tree return true for a given pair, edge is created.
Proposed Method

NIFTY: Phrase Clustering

Phrase Graph Creation : Graph creation optimization

- Brute-Force approach (each pair of phrase)~ $O(n^2)$
- Use Locally-Sensitive-Hashing with min-hashing. Similar items with low Jacquard distance are placed in each bucket.
- Apply edge creation on each pair of phrase in each bucket.
Proposed Method

NIFTY: Phrase Clustering

Phrase Graph Creation: Assigning Edge Weights

- For an edge from node $p_s$ to $p_d$

$$w(p_s, p_d) = c \cdot \frac{|p_d|}{(D_{edit}(p_s, p_d) + 1) \cdot (T_{peak}(p_s, p_d) + 1)}.$$  

Time difference between the first volume peaks for each of the two phrases

Substring edit distance between the phrases
NIFTY: Phrase Clustering

Phrase Graph Creation : Phrase Graph Partitioning

Goal: To delete edges so that each of the components is single rooted

Step 1: Start with working set with all root phrases (zero out degree node)
Step 2: For (nodes not in working set) :
Step 3: If (all the outgoing neighbors are in working set) :
Step 4: Sum the weights of edge with each neighbor assigned to a specific cluster
Step 5: Node is assigned cluster with highest weight

END
END
NIFTY: Phrase Clustering
Phrase Graph Creation : Phrase Graph Partitioning
A number of non-meme clusters such as movies, TV shows etc. are created.

- Filter by phrase mutations
- Filter by peaks: Most news have at most 2 main peaks
NIFTY: Phrase Clustering

Phrase Graph Creation : Incremental Phrase Clustering

We need to update the meme clusters with new stories each day
• Phrase graph creation for new phrase
• Phrase graph partitioning for edges with new phrase
NIFTY: A System for Large Scale Information Flow Tracking and Clustering

Outline
• Introduction
• Goal
• Proposed method
• Experiments
• Conclusions
NIFTY: Evaluation

(a) Running time

(b) Memory usage

Figure 5: MEMETRACKER and NIFTY resource usage.
Figure 6: NIFTY vs. MEMETRACKER cluster size distribution. MEMETRACKER produces a giant cluster of 10,000 phrases.
NIFTY: Evaluation

Figure 7: Basic properties of input data. (a) Number of quotes per document vs. number of such documents. (b) Quote frequency vs. number of such quotes.
NIFTY: A System for Large Scale Information Flow Tracking and Clustering

Outline
- Introduction
- Goal
- Proposed method
- Experiments
- Conclusions
Conclusion

NIFTY has following advantages:

- **Highly Scalable**
  It scales to 6 billion articles
- **Meme mutation dynamics**