Revisit Behavior in Social Media: The Phoenix-R Model and Discoveries

Flavio Figueiredo, Yasuko Matsubara, Bruno Ribeiro,
Jussara M. Almeida, Christos Faloutsos

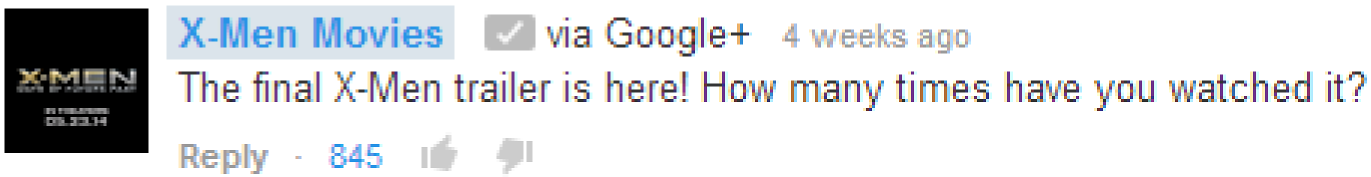
Institute for Web Research (InWeb) @ DCC-UFMG

Databases Group @ CMU

How should we account and model information popularity online?

How should we account and model information popularity online?

Audience: Unique users



Audience vs Visits

Multiple Visits from the Same Users

Measuring both visits and audience (unique users) have their benefits

- How many users watched my ad?
  - Exposure
  - Revenue
- How many times was my ad watched?
  - Caching
  - Sharding and content provisioning
- However…
  - Understanding and modeling both effects is still an open issue

Our Study

- **Understanding** and **modeling** revisit behavior in social media

- **Understanding**
  - Characterization of millions of user activities
  - User played/watched/visited a social media object at a certain time

- **Modeling**
  - The **Phoenix-R** model for popularity time series

Datasets

- User Activity
  - User, Object (song/tweet/video), Time stamp
- All of the datasets range from months to years

Dataset	User Activities	Description
MMTweet (Million Musical Tweets)	Little over 1 million	Tweets declaring songs which users listen to
Twitter	576 million	Hashtags
LastFM	19 million	Plays on artists and songs
YouTube	-	3 million daily time series

Discoveries

Discoveries

- Relationships between audience (unique users) and revisits

Dataset	Median $\frac{\#Revisits}{\#Audience}$	Median $\frac{\#Revisits}{\#Total\ Visits}$	% of cases $\#Revisits > \#Audience$
MMTweet	0.68	0.40	33%
Twitter	1.70	**0.62**	**66%**
LastFM	**25.39**	**0.96**	**100%**

Discoveries on Smaller time Scales

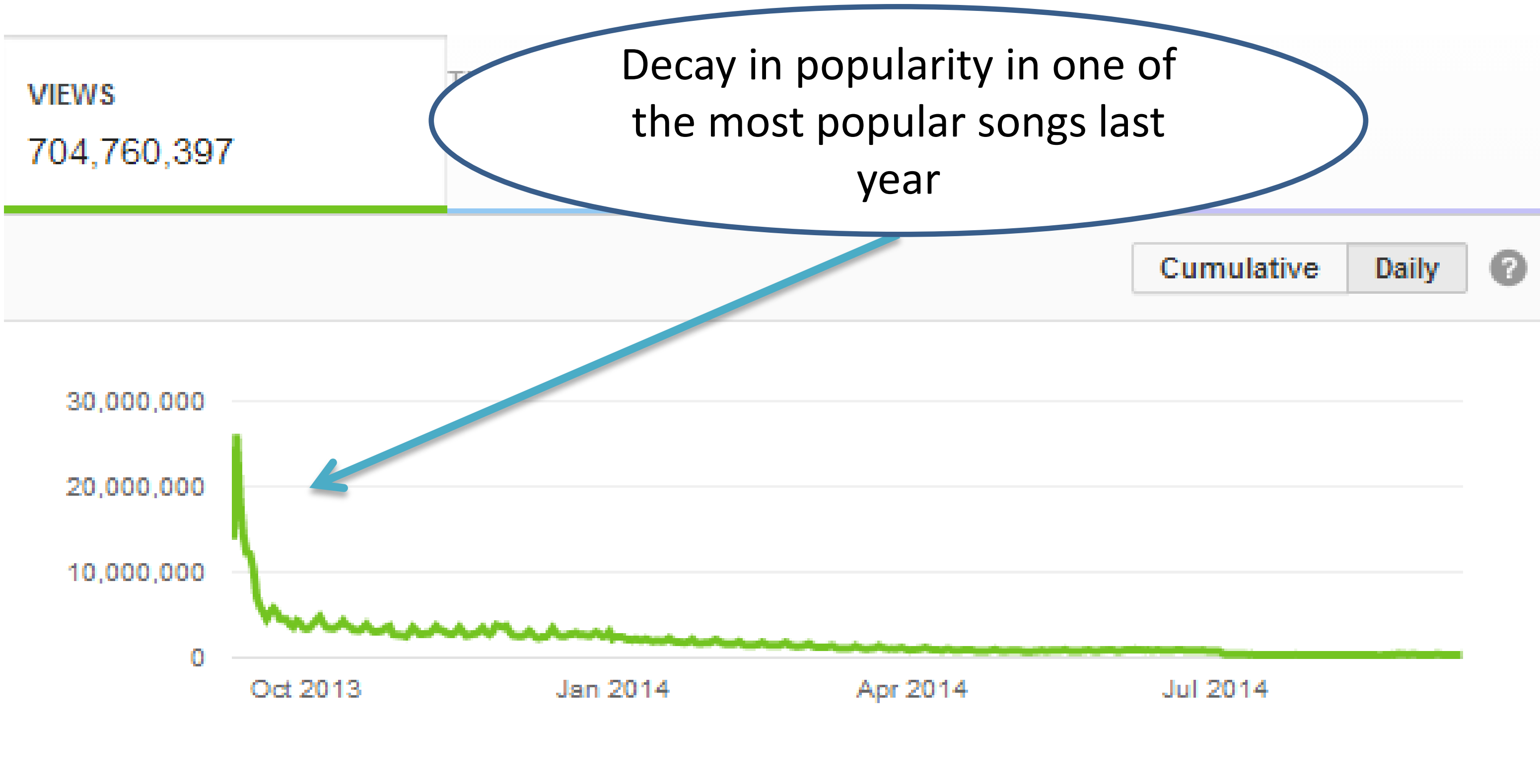
- Isolate the effect of users coming back to the datasets after long periods
- Daily Time Windows

Dataset	Median $\frac{\#Revisits}{\#Audience}$
MMTweet	0.83
Twitter	2.50
LastFM	**28.0**

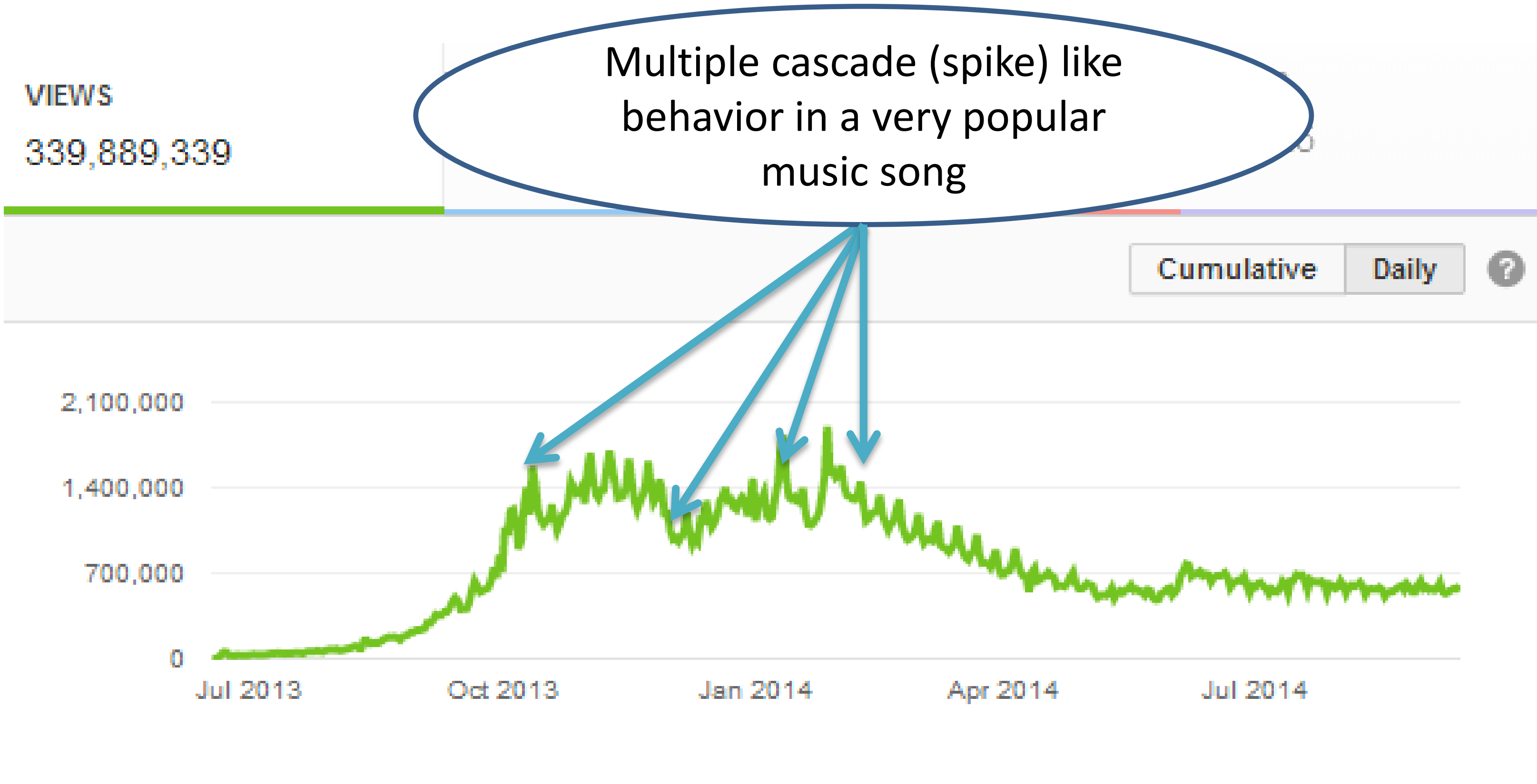
What we know so far

- Users revisit the same object
  - On some datasets (LastFM and Twitter) most of visits are returning users
- Revisits are common on small time scales
  - Above results hold
  - Complements [Anderson2014]
- Users abandon content but it may take a long time
  - Preying behavior from [Ribeiro2014]

Users eventually stop visiting



Some objects behave like a sum of multiple cascades



How de we model these time series?

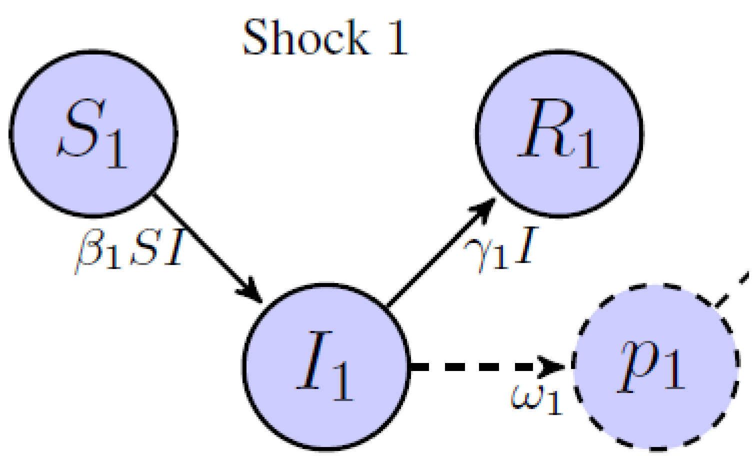
The Phoenix-R Model!

Table 1: Comparison of PHOENIX-R with other approaches

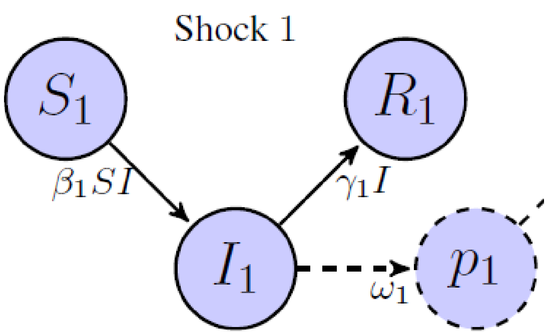
	Revisits	Non-Linear	Forecasting	Multi Cascade
SI [12]		✓		
SpikeM [18]		✓	✓	
TemporalDynamics [21]			✓	
PHOENIX-R	✓	✓	✓	✓

Phoenix-R Explained

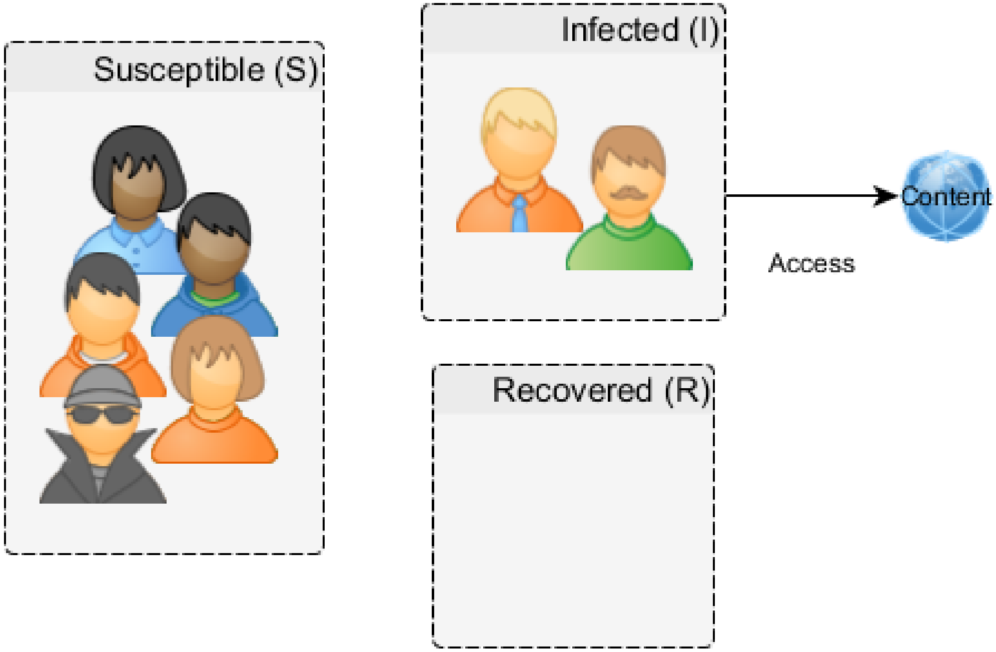
- Single shock (cascade) model
- Epidemiology model



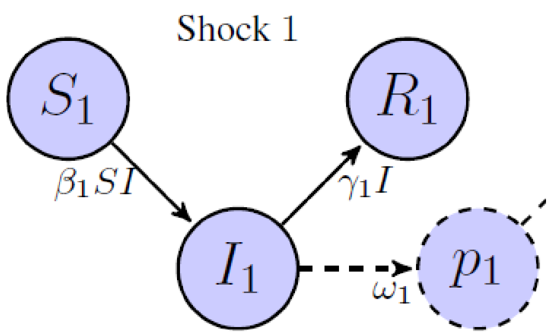
Single Shock



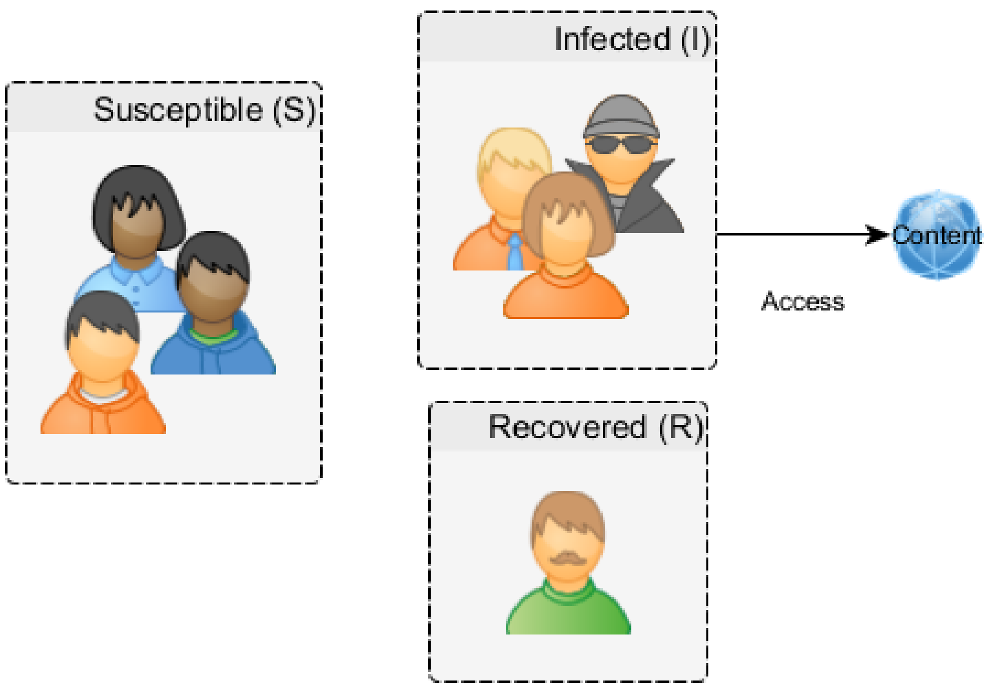
- Starting with some Susceptible and Infected Individuals
- The Infected access the content



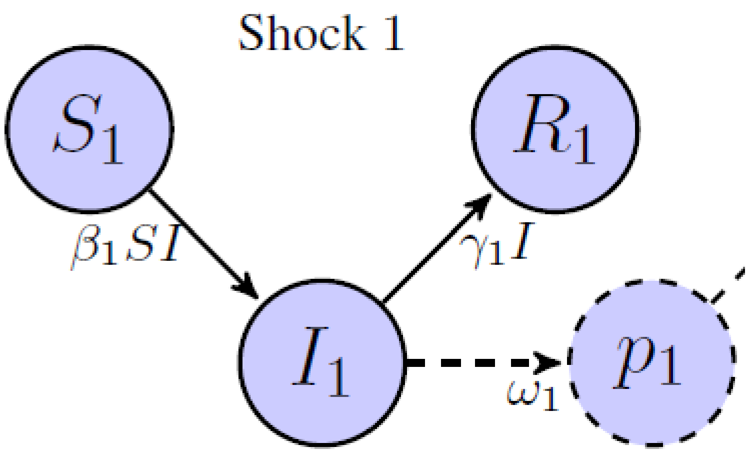
Single Shock



- At the next time tick some Infected recover
- Some Susceptible are infected by the previous infected
- We now expect more visits (more infected)



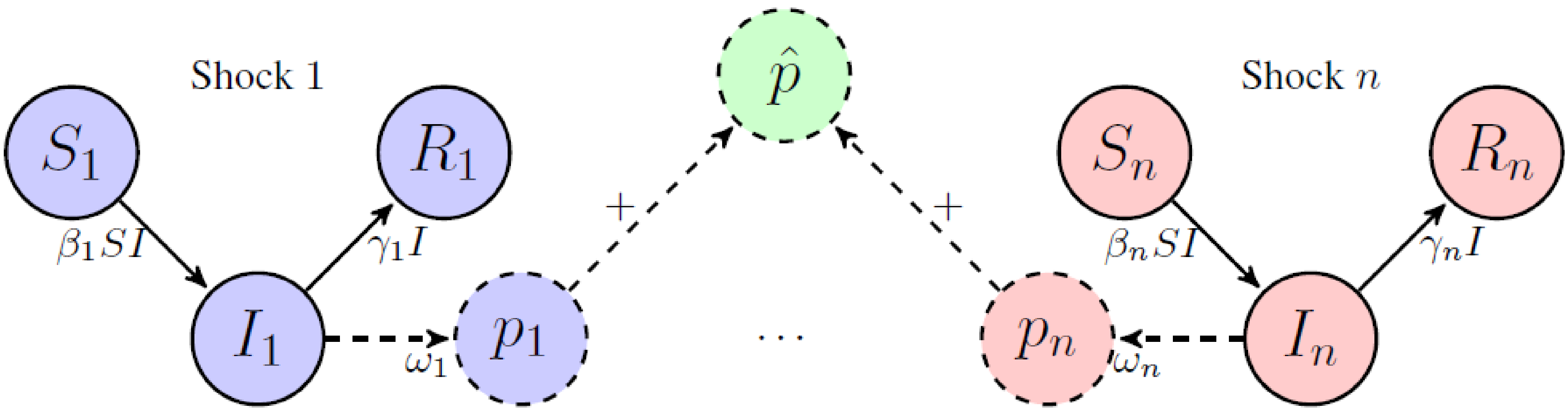
Single Shock Equations



$$S(t) = S(t-1) - \beta S(t-1)I(t-1)$$
$$I(t) = I(t-1) + \beta S(t-1)I(t-1) - \gamma I(t-1)$$
$$R(t) = R(t-1) + \gamma I(t-1)$$
$$p(t) = \omega I(t).$$

Multiple Shocks

- Simplifying assumption that each shock is a new population (set of users)



$$\hat{p}(t) = \sum_{i, s_i \in \mathcal{S}} p_i(t - s_i)\mathbb{1}[t > s_i]$$

How many shocks to add?

- A perfect model (zero error) can be created by
  - Letting each access be a single user which immediately recovers
  - However, lot's of parameters (cost)

- Using Minimum Description Length (MDL)

$$Cost(\mathbf{t}; \mathcal{P}) = \log^* n + Cost(\mathcal{P}) + Cost(\mathbf{t} \mid \mathcal{P})$$

How do we fit a time series?

- Step 1:
  - Identify Peaks using Wavelets
  - Intuitively, each peak is a candidate shock (cascade)
  - Linear
- Step 2:
  - Add each peak sorted by height to the model
  - If the MDL decreases, accept peak
- Step 3:
  - Stop when the MDL stops decreasing

Linear runtime (time series length) and parameter free algorithm

Algorithm 1 Fitting the PHOENIX-R model. Only the time series is required as input.

function FITPHOENIXR($\mathbf{t}$)
    $\epsilon = 0.05$
    $\mathbf{s} \leftarrow \{\}$
    $\mathbf{p}, \mathbf{s}' \leftarrow FindPeaks(\mathbf{t})$ Find Peaks
    $\mathbf{s}[1] = 0$
    $\mathbf{s} \leftarrow append(\mathbf{s}')$
    $\mathcal{P} \leftarrow \{\}$
    $min\_cost \leftarrow \infty$
    for $i \leftarrow 1$ to $|s|$ **do**
        $\mathcal{F} \leftarrow LM(\mathbf{t}, \mathbf{s}(:i))$
        $\mathbf{m} \leftarrow PhoenixR(\mathcal{F})$ Adding shocks
        $mdl\_cost \leftarrow Cost(\mathbf{m}, \mathbf{t}, \mathcal{F})$
        if $mdl\_cost < min\_cost$ **then**
            $min\_cost \leftarrow mdl\_cost$
            $\mathcal{P} \leftarrow \mathcal{F}$
        end if
        if $mdl\_cost > min\_cost * (1 + \epsilon)$ **then**   Exit
            break
        end if
    end for
    return $\mathcal{P}$
end function

How good is Phoenix-R?

- Comparing Phoenix-R with two state of the art alternatives
  - RMSE (smaller is better)

	PHOENIX-R vs. TemporalDynamics (daily series)		PHOENIX-R vs. SpikeM (hourly series)	
	RMSE PHOENIX-R	RMSE TemporalDynamics	RMSE PHOENIX-R	RMSE SpikeM
MMTweet	**2.93** (± 0.23)	4.18 (± 0.49)	-	-
LastFM	**7.09** (± 0.23)	8.31 (± 0.32)	-	-
Twitter	**72.05** (± 6.08)	194.79 (± 20.49)	**10.83** (± 1.61)	**9.77** (± 2.24)
YouTube	**280.58** (± 29.29)	3429.19 (± 577.76)	-	-

- Phoenix-R is always better or just as good

How good is Phoenix-R?

- Comparing Phoenix-R with two state of the art alternatives
  - RMSE (smaller is better)

	PHOENIX-R vs. TemporalDynamics (daily series)		PHOENIX-R vs. SpikeM (hourly series)	
	RMSE PHOENIX-R	RMSE TemporalDynamics	RMSE PHOENIX-R	RMSE SpikeM
MMTweet	**2.93** (± 0.23)	4.18 (± 0.49)	-	-
LastFM	**7.09** (± 0.23)	8.31 (± 0.32)	-	-
Twitter	**72.05** (± 6.08)	194.79 (± 20.49)	**10.83** (± 1.61)	**9.77** (± 2.24)
YouTube	**280.58** (± 29.29)	3429.19 (± 577.76)	-	-

- Phoenix-R is always better or just as good

Phoenix-R is also good at forecasting

- RMSE (smaller is better)
- 1, 7 or 30 days ahead forecasting
- Ties on very linear time series

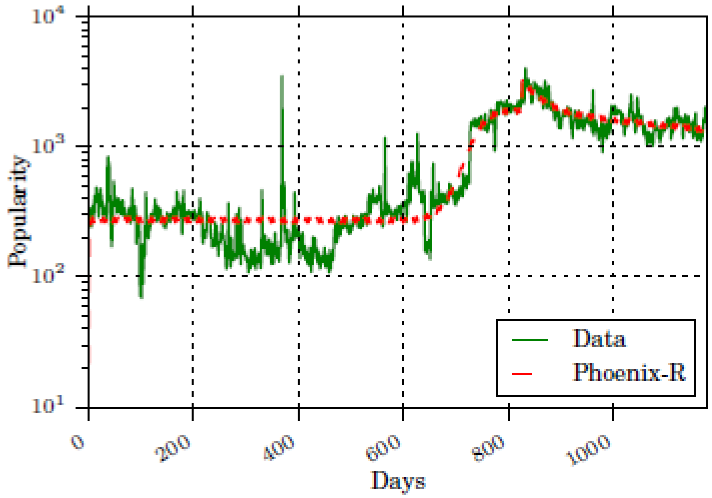
		5%			25%			50%		
		1	7	30	1	7	30	1	7	30
MMTweet	PhoenixR	**11.61**	**12.78**	**15.15**	**8.67**	**6.74**	**8.82**	**4.08**	**6.87**	**13.58**
	TempDynamics	**17.07**	**17.41**	**16.52**	**9.63**	**10.78**	**14.46**	**25.19**	**23.08**	**30.39**
Twitter	PhoenixR	**53.68**	**60.78**	**215.76**	**132.21**	**135.15**	**210.30**	**75.58**	**229.59**	**254.93**
	TempDynamics	**104.45**	**129.36**	**255.69**	643.39	643.83	786.50	420.74	587.86	598.75
LastFM	PhoenixR	**2.37**	**3.97**	**5.71**	**8.60**	**12.06**	**14.66**	**11.34**	**15.03**	**15.43**
	TempDynamics	6.47	7.03	8.00	11.15	14.62	17.86	14.91	18.15	18.80
YouTube	PhoenixR	**91.62**	**106.38**	**138.88**	**83.76**	**113.14**	**147.04**	**127.53**	**97.97**	**115.97**
	TempDynamics	3560.65	3631.09	3661.81	5091.82	5107.82	5143.70	4136.14	4139.73	4169.26

Phoenix-R is also good at forecasting

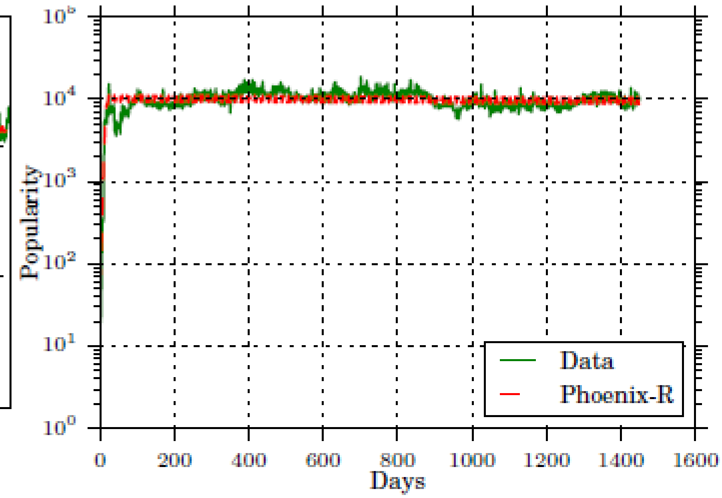
- RMSE (smaller is better)
- 1, 7 or 30 days ahead forecasting
- Ties on very linear time series

		5%			25%			50%		
		1	7	30	1	7	30	1	7	30
MMTweet	PhoenixR	**11.61**	**12.78**	**15.15**	**8.67**	**6.74**	**8.82**	**4.08**	**6.87**	**13.58**
	TempDynamics	**17.07**	**17.41**	**16.52**	**9.63**	**10.78**	**14.46**	**25.19**	**23.08**	**30.39**
Twitter	PhoenixR	**53.68**	**60.78**	**215.76**	**132.21**	**135.15**	**210.30**	**75.58**	**229.59**	**254.93**
	TempDynamics	**104.45**	**129.36**	**255.69**	643.39	643.83	786.50	420.74	587.86	598.75
LastFM	PhoenixR	**2.37**	**3.97**	**5.71**	**8.60**	**12.06**	**14.66**	**11.34**	**15.03**	**15.43**
	TempDynamics	6.47	7.03	8.00	11.15	14.62	17.86	14.91	18.15	18.80
YouTube	PhoenixR	**91.62**	**106.38**	**138.88**	**83.76**	**113.14**	**147.04**	**127.53**	**97.97**	**115.97**
	TempDynamics	3560.65	3631.09	3661.81	5091.82	5107.82	5143.70	4136.14	4139.73	4169.26

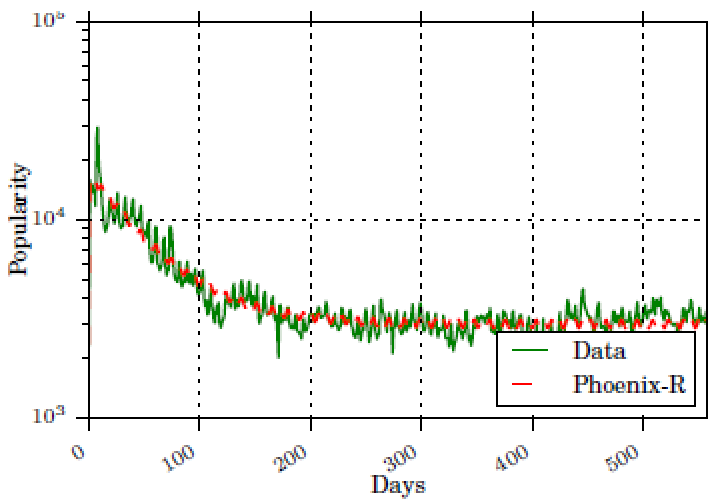
Examples of Phoenix-R at work



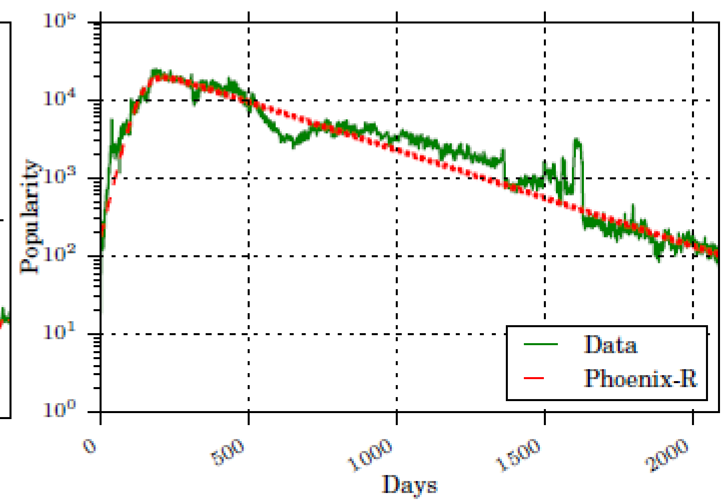
(a) Rock Song (growth in popularity)



(b) Flashdance (80's movie) clip (revisits)

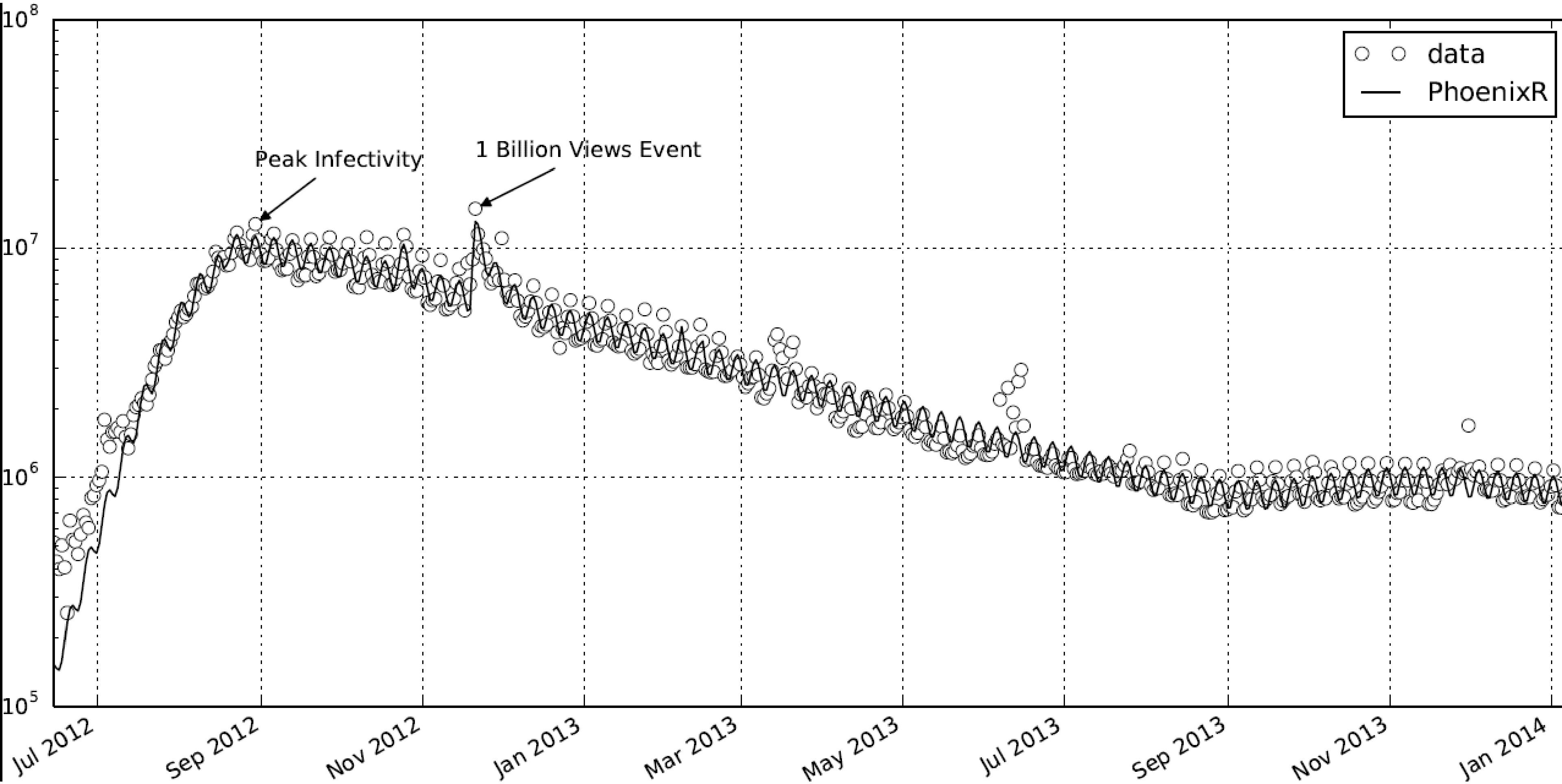


(c) Korean Music Video (single cascade)



(d) User Dancing Video (single cascade)

Examples of Phoenix-R at work



Conclusions

- Phoenix-R model for revisits and multiple cascades
- Based on discoveries from real data
- Scalable linear fitting algorithm
  - On time series length
- Useful for predictions