Vote-and-Comment: Modeling the Coevolution of User Interactions in Social Voting Web Sites

Alceu Ferraz Costa[1] Agma Juci Machado Traina[1]
Caetano Traina Jr. [1] Christos Faloutsos[2]

IEEE International Conference on Data Mining

[1]University of São Paulo

[2]Carnegie Mellon University

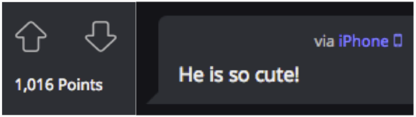
Outline

Social Voting Web Sites

In social voting web sites, users can submit content (e.g. pictures, news articles)

And other users can:

- Up-vote (like)
- Down-vote (dislike)
- Post comments



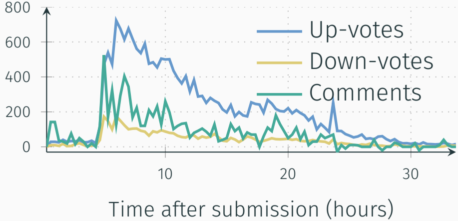
Examples of social voting web sites: Reddit, Imgur, Hacker-News

Coevolution of User Interactions

For a submission, we have 3 time-series: up-votes $v_+(t)$, down-votes $v_-(t)$ and comments $c(t)$:

Submission

Time-Series



Problem

Can we explain how $v_+(t)$, $v_-(t)$ and $c(t)$ evolve over time?

Outline

Vote-and-Comment (VnC) Model

VnC: mathematical model that describes how the volume of up-votes, down-votes and comments changes over time

VnC is composed of 3 submodels that describe the following relationships:

1. Up-votes over time
2. Up-votes vs. down-votes
3. Comments vs. votes

VnC: Up-votes Over Time

The number $v_+(t)$ of up-votes received by a submission at time t is a function of:

1. $P(t)$: probability of a user up-voting at time t
2. N_+: population of potential voters
3. $V_+(t)$: number of votes *accumulated* at time t

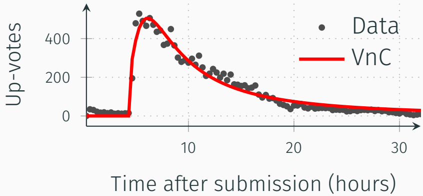
VnC: Up-votes Over Time

The number $v_+(t)$ of up-votes received by a submission at time t is a function of:

1. $P(t)$: probability of a user up-voting at time t
2. N_+: population of potential voters
3. $V_+(t)$: number of votes *accumulated* at time t

Up-votes over Time

$$v_+(t+1) = \underbrace{[N_+ - V_+(t)]}_{\text{users that can vote}} \cdot P(t)$$

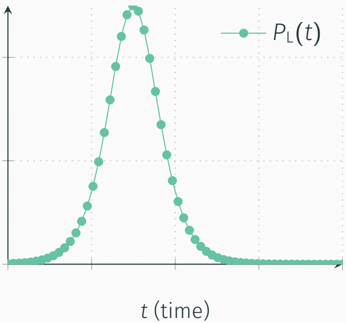


Next Step: How can we model the probability $P(t)$?

VnC: Modeling the Up-voting Probability

$P_{\mathrm{L}}(t;\beta_+,\xi_+)$: Probability that a user likes a submission

- **Cascading Mechanism:** submission popularity affects voting probability
- $P_{\mathrm{L}}(t) = \xi_+ + \beta_+ \cdot V_+(t)/N_+$



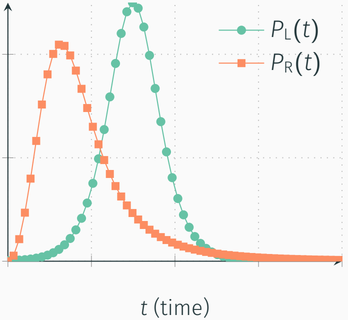
VnC: Modeling the Up-voting Probability

$P_{\mathrm{L}}(t; \beta_+, \xi_+)$: Probability that a user likes a submission

- **Cascading Mechanism:** submission popularity affects voting probability
- $P_{\mathrm{L}}(t) = \xi_+ + \beta_+ \cdot V_+(t)/N_+$

$P_{\mathrm{R}}(t; \mu, s)$: Probability that a user reacts at time t

- Log-logistic with parameters μ and s Details



VnC: Modeling the Up-voting Probability

$P_{\mathrm{L}}(t; \beta_+, \xi_+)$: Probability that a user likes a submission

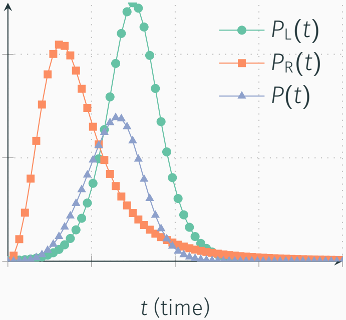
- **Cascading Mechanism:** submission popularity affects voting probability
- $P_{\mathrm{L}}(t) = \xi_+ + \beta_+ \cdot V_+(t)/N_+$

$P_{\mathrm{R}}(t; \mu, s)$: Probability that a user reacts at time t

- Log-logistic with parameters μ and s Details

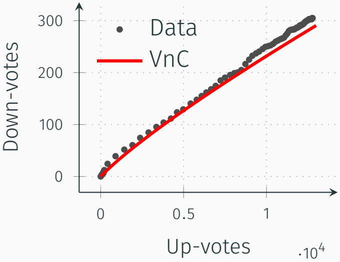
$P(t)$: Probability of a user up-voting at time t

- $P(t) = P_{\mathrm{L}}(t) \cdot P_{\mathrm{R}}(t)$



VnC: Up-votes vs. Down-votes

$$v_{-}(t+1) = [N_{-} - V_{-}(t)] \cdot P_{\mathrm{L}}(t; \beta_{-}, \xi_{-}) \cdot P_{\mathrm{R}}(t; \mu, s)$$

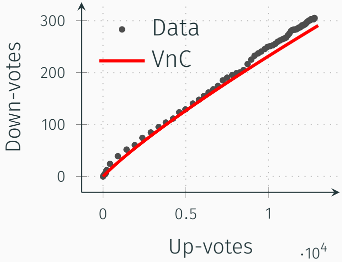


VnC: Up-votes vs. Down-votes

$$v_{-}(t+1) = [N_{-} - V_{-}(t)] \cdot \overbrace{P_{\mathrm{L}}(t;\beta_{-},\xi_{-})}^{\text{Cascading}} \cdot \overbrace{P_{\mathrm{R}}(t;\mu,s)}^{\text{Reaction Times}}$$

The down-vote time-series $v_{-}(t)$ also follows:

1. A cascading mechanism
2. Log-logistic reaction times

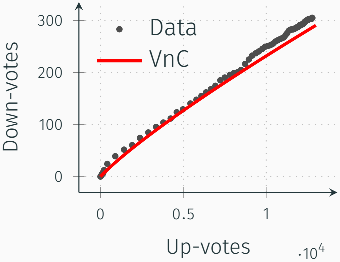


VnC: Up-votes vs. Down-votes

$$v_{-}(t+1) = \underbrace{[N_{-} - V_{-}(t)] \cdot \overbrace{P_{\mathrm{L}}(t;\beta_{-},\xi_{-})}^{\text{Cascading}}}_{\text{Not-shared Parameters}} \cdot \underbrace{\overbrace{P_{\mathrm{R}}(t;\mu,s)}^{\text{Reaction Times}}}_{\text{Shared}}$$

The down-vote time-series $v_{-}(t)$ also follows:

1. A cascading mechanism
2. Log-logistic reaction times



Sharing parameters with the up-vote time-series:

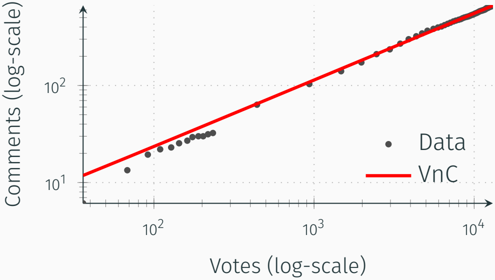
1. Shared: μ and s
2. Not shared: N_{-}, β_{-} and ξ_{-}

VnC: Comments vs. Votes

VnC models the number of comments $C(t)$ as a power law on the number of votes:

$$C(t) = k \cdot [V_+(t) + V_-(t)]^\alpha$$

The power-law —— matches the data



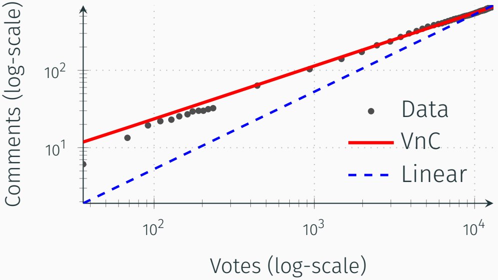
VnC: Comments vs. Votes

VnC models the number of comments $C(t)$ as a power law on the number of votes:

$$C(t) = k \cdot [V_+(t) + V_-(t)]^\alpha$$

The power-law —— matches the data

- The linear relationship fails to match the data



Outline

Experiments: Questions

Our goal is to answer the following questions:

- **Q1 – Fit Accuracy:** Is VNC more accurate than existing models when fitting social voting data?
- **Q2 – Popularity Decay:** Can VNC model the popularity decay of submissions?
- **Q3 – Coevolution:** Can VNC model the coevolution of up-votes, down-votes and comments time-series?

Datasets

Our crawler tracked Reddit and Imgur submissions:

- Collected the number of votes and comments every 20 minutes
- Submissions were tracked for 33 hours after their creation
- Submissions with less than 100 up-votes were discarded

Digg dataset publicly avaiable (Lerman and Ghosh, 2010):

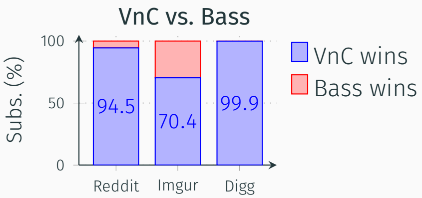
- Only up-votes (no down-votes and comments data)

Dataset	# Submissions	# User Interactions
Reddit	17,205	113,331,266
Imgur	724	2,107,576
Digg	3,553	5,149,170

Q1 – Fit Accuracy

Percentage of up-vote time-series that were best fit by each model

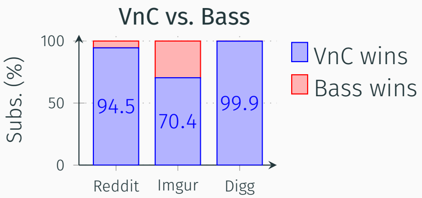
- Best fit determined by smaller root-mean-square error Details

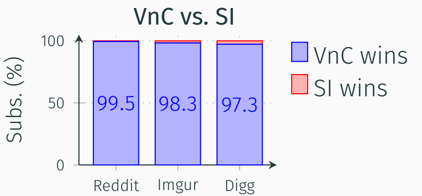


Q1 – Fit Accuracy

Percentage of up-vote time-series that were best fit by each model

- Best fit determined by smaller root-mean-square error Details

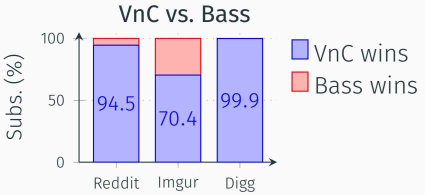


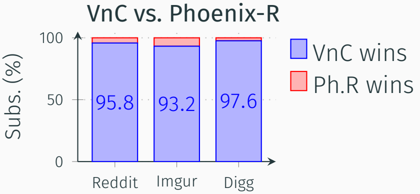


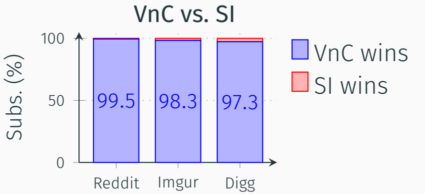
Q1 – Fit Accuracy

Percentage of up-vote time-series that were best fit by each model

- Best fit determined by smaller root-mean-square error Details



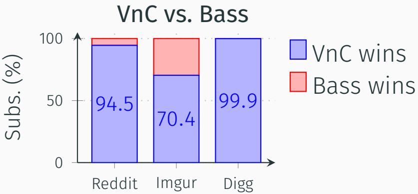


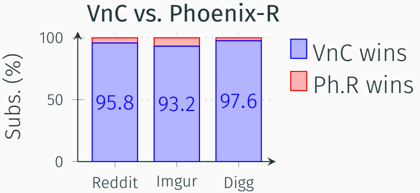


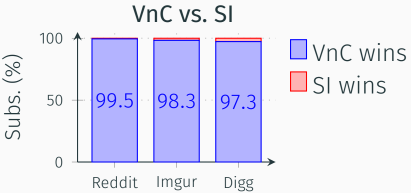
Q1 – Fit Accuracy

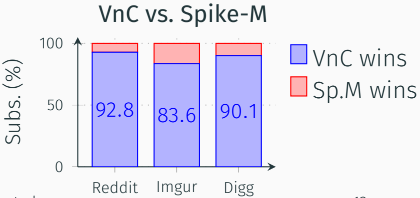
Percentage of up-vote time-series that were best fit by each model

- Best fit determined by smaller root-mean-square error Details



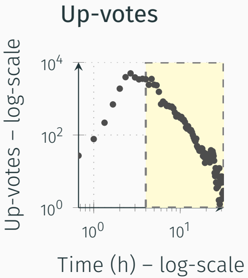


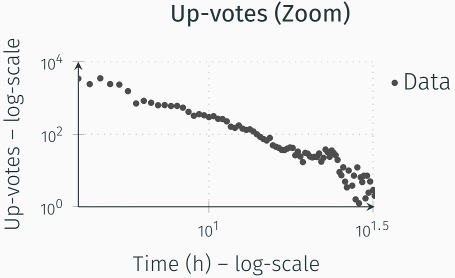




Q2 – Popularity Decay

Up-vote time-series have a heavy-tail decay

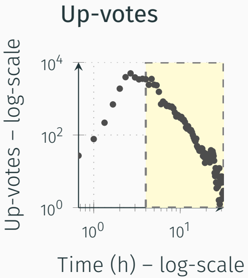


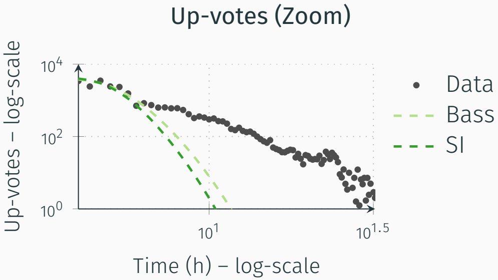


Q2 – Popularity Decay

Up-vote time-series have a heavy-tail decay

Bass and SI models generate unrealistic exponential decays

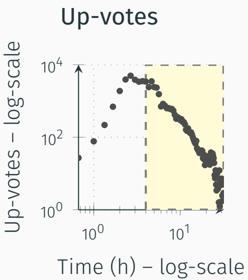


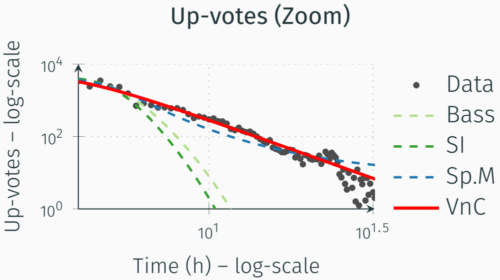


Up-vote time-series have a heavy-tail decay

Bass and SI models generate unrealistic exponential decays

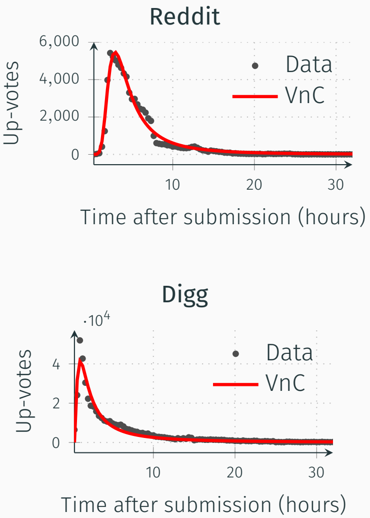
VnC and Spike-M are able to match the heavy tail decay





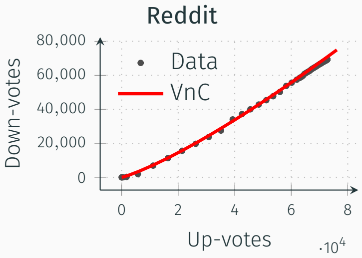
Q3 – Coevolution: Up-votes over Time

VnC up-vote time-series fits for the most voted submissions in each dataset

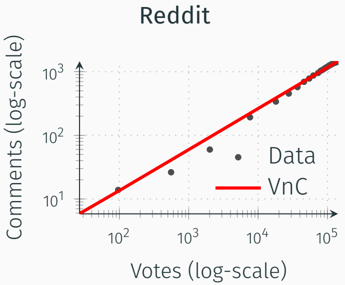


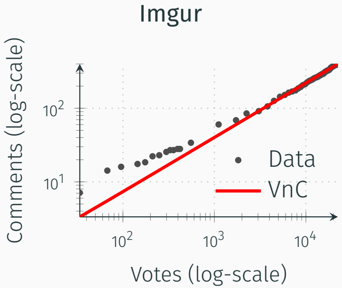
Q3 – Coevolution: Up-votes vs. Down-votes

VNC fit for the relationship between up-votes and down-votes received by a submission



VnC fit for the relationship between total votes (up-votes + down-votes) and comments received by a submission





Outline

1. Introduction

2. Model

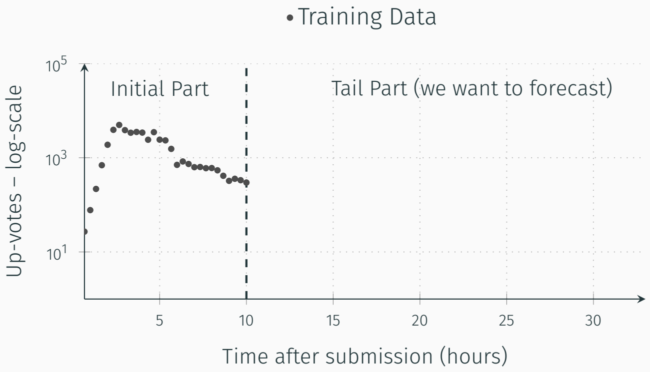
3. Experiments

4. VnC: Applications

5. Conclusions

Forecasting

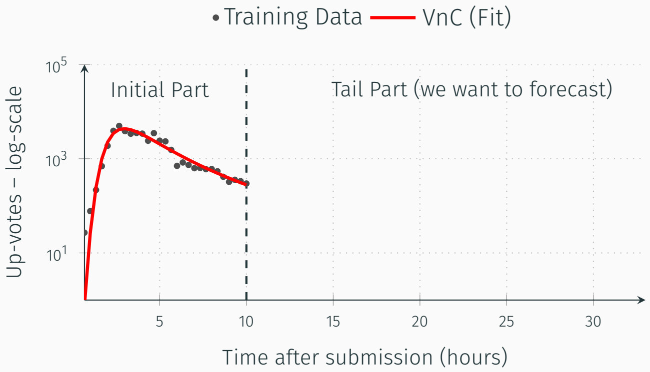
Problem: Given the initial part of a social voting time-series, predict the tail part



Forecasting

Problem: Given the initial part of a social voting time-series, predict the tail part

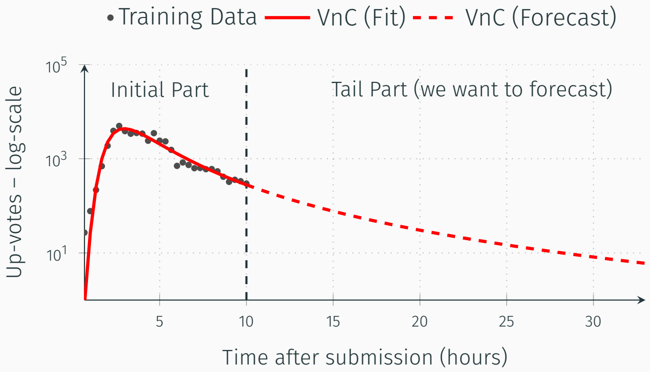
1. Estimate VnC parameters using the initial part



Forecasting

Problem: Given the initial part of a social voting time-series, predict the tail part

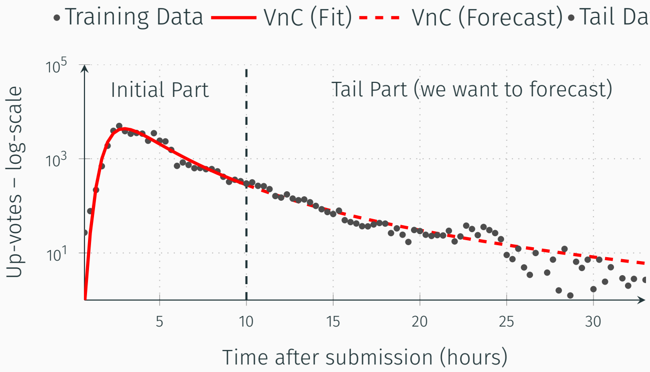
1. Estimate VnC parameters using the initial part
2. Use the parameters to forecast the tail part



Forecasting

Problem: Given the initial part of a social voting time-series, predict the tail part

1. Estimate VnC parameters using the initial part
2. Use the parameters to forecast the tail part

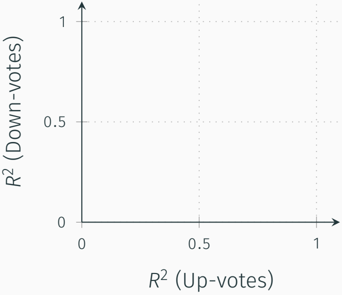


Outlier Detection

To detect outliers we use VNC's R^2

- R^2 measures fit accuracy
- R^2 values closer to 1 indicate better fits

R^2 vs. R^2 plot:



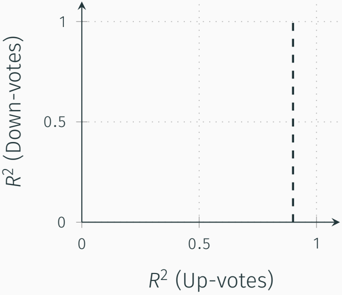
Outlier Detection

To detect outliers we use VNC's R^2

- R^2 measures fit accuracy
- R^2 values closer to 1 indicate better fits

R^2 vs. R^2 plot:

1. Compute R^2 for the up-vote time-series



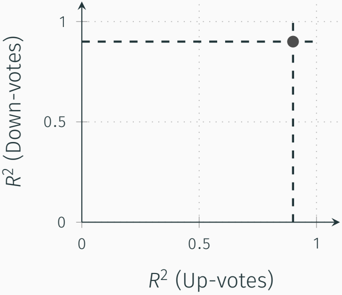
Outlier Detection

To detect outliers we use VNC's R^2

- R^2 measures fit accuracy
- R^2 values closer to 1 indicate better fits

R^2 vs. R^2 plot:

1. Compute R^2 for the up-vote time-series
2. Compute R^2 for the down-vote time-series



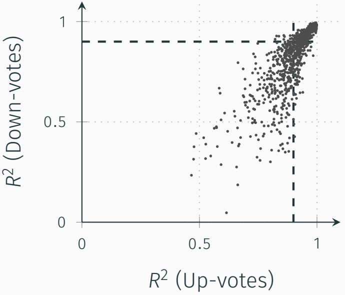
Outlier Detection

To detect outliers we use VNC's R^2

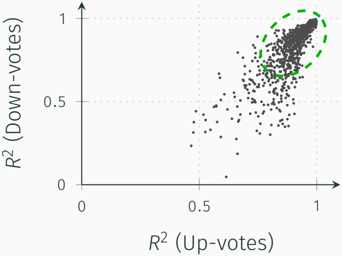
- R^2 measures fit accuracy
- R^2 values closer to 1 indicate better fits

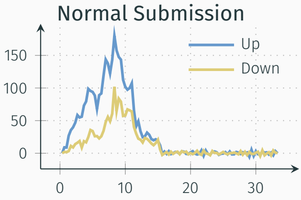
R^2 vs. R^2 plot:

1. Compute R^2 for the up-vote time-series
2. Compute R^2 for the down-vote time-series
3. Repeat for all submissions



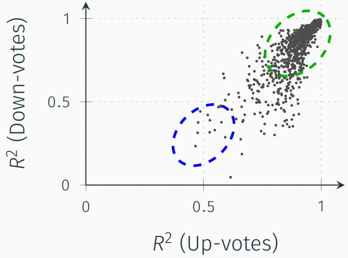
Outlier Detection

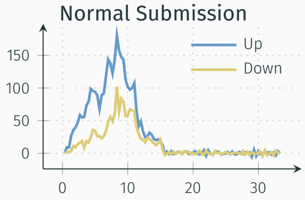




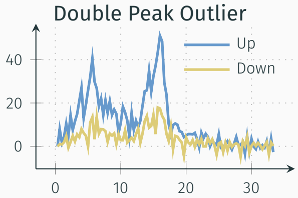
Over 95% of the submissions

Outlier Detection



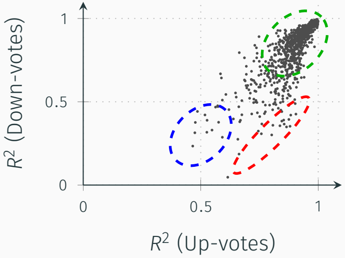


Over 95% of the submissions

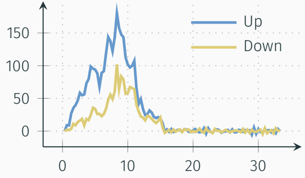


Late-night submissions: 1st peak at night, 2nd peak at morning

Outlier Detection

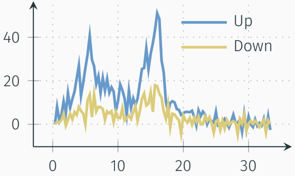


Normal Submission



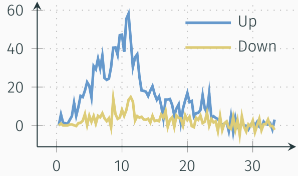
Over 95% of the submissions

Double Peak Outlier

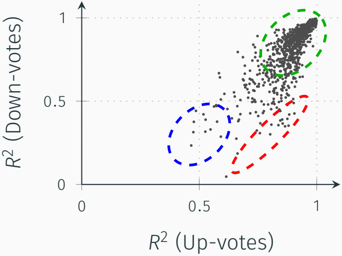


Late-night submissions: 1st peak at night, 2nd peak at morning

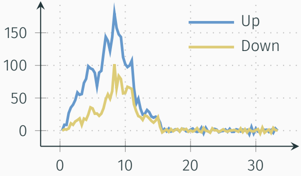
Flat Down-vote Outlier



Most are pictures of animals

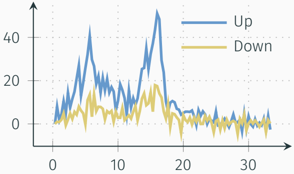


Normal Submission



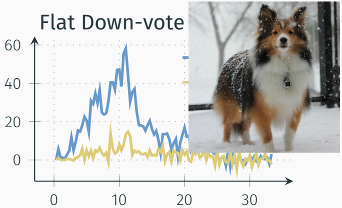
Over 95% of the submissions

Double Peak Outlier



Late-night submissions: 1st peak at night, 2nd peak at morning

Flat Down-vote



Most are pictures of animals

Outline

Conclusions

VNC has the following advantages:

- Coevolution Modeling: Describes up-vote, down-vote and comments time-series
- Praticality: Matches data from several social voting web sites
- Usefulness: Forecasting and outlier detection

Questions?

Vote-and-Comment: Modeling the Coevolution of User Interactions in Social Voting Web Sites

Alceu Ferraz Costa
Agma Juci Machado Traina
Caetano Traina Jr.
Christos Faloutsos

Dataset and Code: **`https://github.com/alceufc/vnc_model`**

Email: alceufc@icmc.usp.br

Funding:

Reaction Probability (Details)

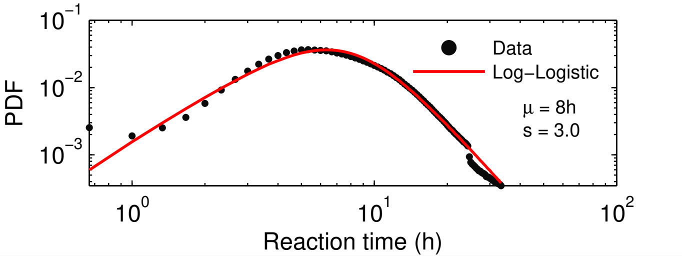
Definition (Reaction Time)

The reaction time corresponds to the time interval between the instant in which a submission is created and the instant in which an interaction (vote or comment) occurs.

Data: Reaction times of up-votes, down-votes and comments from the Reddit, Imgur and Digg datasets.

Log-Logistic PDF

$$f(x) = \frac{(s/\mu)(x/\mu)^{s-1}}{(1+(x/\mu)^s)^2}$$



Parameter Estimation (Details)

To fit the up-vote time-series, VNC uses a set of 6 parameters denoted by $\theta = \{N, \beta_+, \xi_+, \mu, s, t_s\}$:

- N_+: Population
- β_+ and ξ_+: Cascading and independent coefficients
- μ and s: Log-logistic scale and shape parameters
- t_s: Share time

Given:

- A time-series $v_+(t)$ of real data
- The VNC estimated values $\hat{v_+}(t;\theta)$

We learn the parameters by minimizing the sum of squared errors between:

$$\min_{\theta} \sum_{t=1}^{N} [v_+(t) - \hat{v_+}(t;\theta)]^2$$

Delayed Up-voting

Some social voting Web sites allow users to create a submission at time $t = 1$ but only share it later at time $t = t_s$

VNC models this as follows:

- $P(t) = 0$ if $t \leq t_s$
- $P(t) = P_L(t) \cdot P_R(t - t_s)$ if $t > t_s$

Forecasting (Details)

APE: Absolute Percentage Error

- $\text{APE} = |A - F|/A$, where:
- A: actual number of up-votes, down-votes or comments
- F: forecasted number of up-votes, down-votes or comments

Table 1: Median forecasting APE.

		Bass	SI	Phoenix-R	Spike-M	**VnC**
Reddit	v_+	0.57	0.57	0.82	0.42	**0.39**
	v_-	0.67	0.64	0.86	**0.55**	**0.53**
	c	0.62	0.64	0.86	0.73	**0.39**
Imgur	v_+	**0.69**	**0.65**	0.81	0.98	**0.71**
	v_-	**0.65**	0.68	0.84	0.98	**0.65**
	c	0.59	0.58	0.84	1.15	**0.47**
Digg	v_+	0.87	0.89	0.98	**0.52**	0.77

Fit Accuracy (Details)

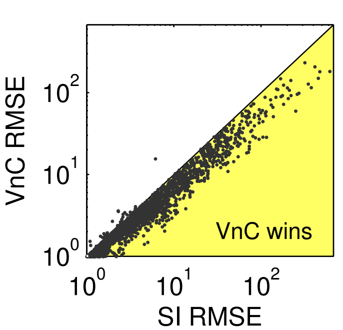
Root-Mean-Square Error

- Lower RMSE indicates a better fit
- Points below the diagonal: time-series that were best fit by VnC

Is VnC more accurate than:

- SI Model? **Yes**
- Bass Model?
- Phoenix-R?
- Spike-M?

VnC vs. SI Model



VnC is more accurate than **SI model** for **99%** of the submissions

Fit Accuracy (Details)

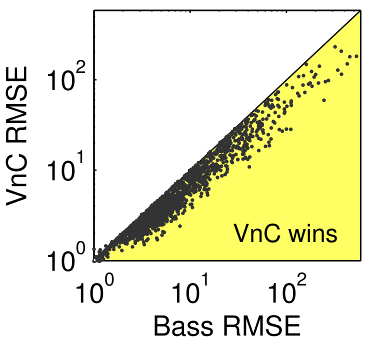
Root-Mean-Square Error

- Lower RMSE indicates a better fit
- Points below the diagonal: time-series that were best fit by VnC

Is VnC more accurate than:

- SI Model? **Yes**
- Bass Model? **Yes**
- Phoenix-R?
- Spike-M?

VnC vs. Bass Model



VnC is more accurate than **Bass Model** for **91%** of the submissions

Fit Accuracy (Details)

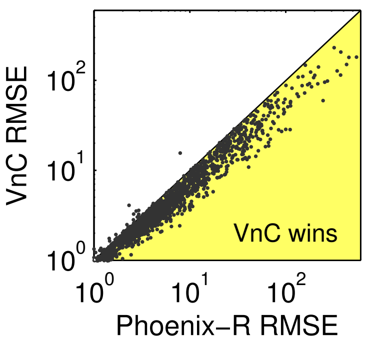
Root-Mean-Square Error

- Lower RMSE indicates a better fit
- Points below the diagonal: time-series that were best fit by VnC

Is VnC more accurate than:

- SI Model? **Yes**
- Bass Model? **Yes**
- Phoenix-R? **Yes**
- Spike-M?

VnC vs. Phoenix-R



VnC is more accurate than **Phoenix-R** for **96%** of the submissions

Fit Accuracy (Details)

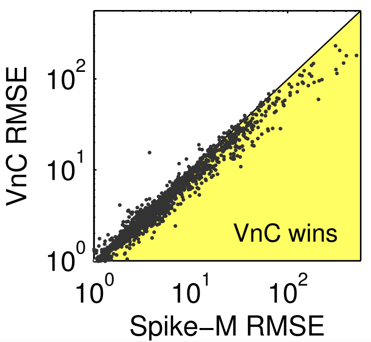
Root-Mean-Square Error

- Lower RMSE indicates a better fit
- Points below the diagonal: time-series that were best fit by VnC

Is VnC more accurate than:

- SI Model? **Yes**
- Bass Model? **Yes**
- Phoenix-R? **Yes**
- Spike-M? **Yes**

VnC vs. Spike-M



VnC is more accurate than **Spike-M** for **90%** of the submissions