The TROOTH Recommendation System

Keno Albrecht

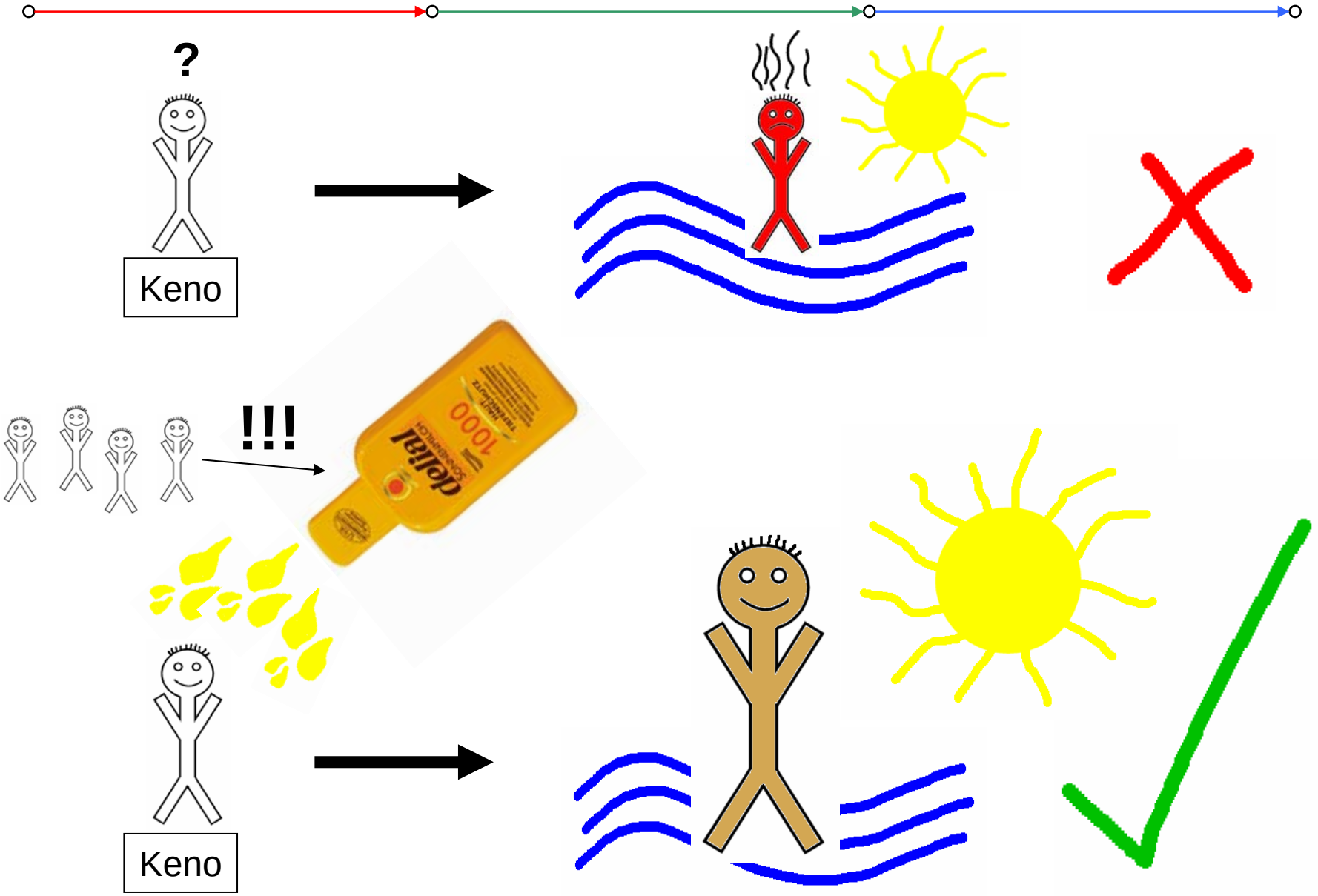
Roger Wattenhofer

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

Motivation



Motivation

- Taking advice of friends makes “vacation” more pleasant.
- In general: Listening to somebody with more experiences is a great idea.
- But listening to strangers also bears some problems: beware of liars!

Recommendation of Books

Customers who bought this also bought

Eldest (Inheritance, Book 2) by *Christopher Paolini*

Harry Potter Paperback Boxed Set (Books 1-5) by *J. K. Rowling*

Harry Potter and the Goblet of Fire (Book 4) by *J.K. Rowling*

Harry Potter and the Prisoner of Azkaban (Book 3) by *J.K. Rowling*

The Opal Deception (Artemis Fowl, Book 4) by *Eoin Colfer*

Explore similar items: in Books, in DVD

[© Amazon.com]

Reputation of Users

Seller information

the_antiquarium (136 ★) me

Feedback Score: 136
Positive Feedback: 100%
Member since Apr-12-00 in United States

Read feedback comments

Add to Favorite Sellers

Ask seller a question

View seller's other items

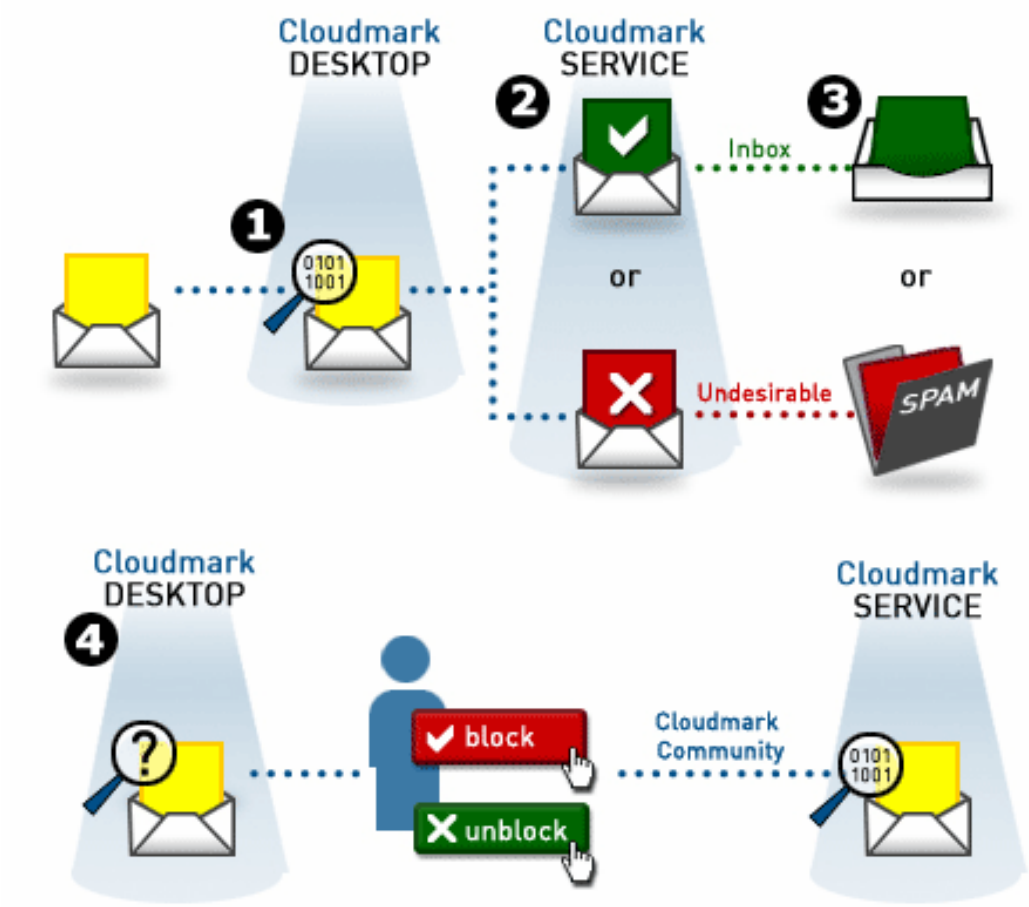
[© Ebay.com]

Rating of Software

[© Cnet.com]

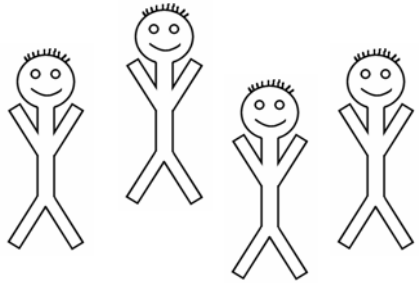
Popularity of Topics & Web Pages

Legitimacy of Emails



[© Cloudmark.com]

Recommendation Systems



Learn from people’s experiences.

Different Flavors of Voting

- What? Users & “items”
- How? Implicit & explicit voting
- Why? Recommendation, rating, trust
- Where? Server-side vs. client-side data

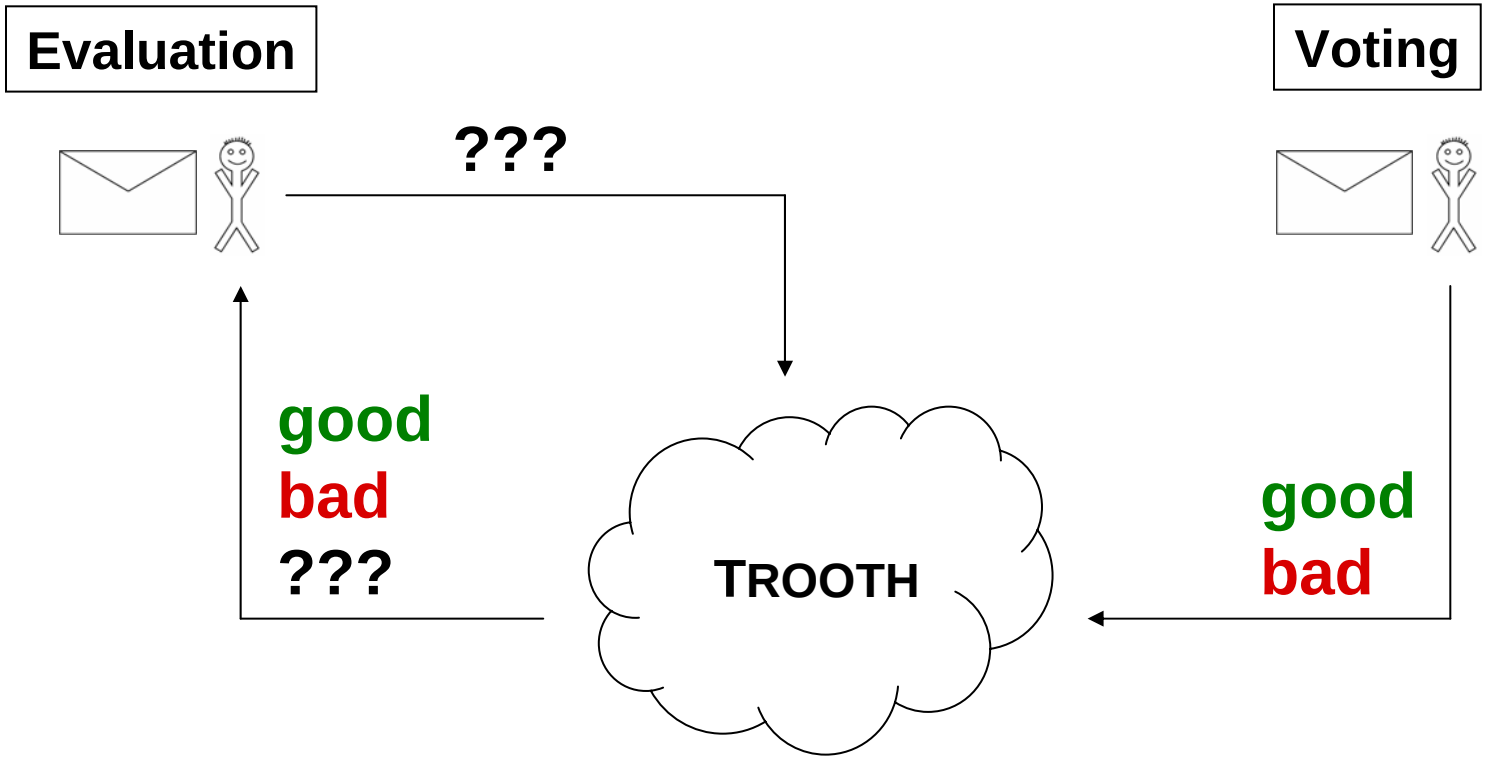
→ Derive future decisions from past ones

TROOTH

- Implicitly create trust values for users by explicitly rating items.
- Store few data on servers; evaluate items on clients.
- Derive decision about items by considering ratings of most trusted users.

Overview

- Items are either *good*, *bad*, or *unknown*.
- Users classify items to be *good* or *bad.*



Simple Evaluation

- If a majority of all users votes *good* (*bad*), the item is *good* (*bad*), otherwise unknown.

- More general:

$$eval(Votes) = \begin{cases} good & \text{if } h_g < \rho_g \leq 1, \\ bad & \text{if } 0 \leq \rho_g < h_b, \\ unknown & \text{if } h_b \leq \rho_g \leq h_g. \end{cases}$$

Weighted Evaluation

- Consider users with different trust values, separating them into *good* and *bad* users.
- Weight votes with trust values before evaluating item.

- Additive Increase, Multiple Decrease:

$$\forall u \in U^i : \quad t'_u := \begin{cases} t_u + inc & \text{if } v_u^i = e^i, \\ t_u \cdot dec & \text{if } v_u^i \neq e^i. \end{cases}$$

TROOTH - Assumptions

- Evaluation of items is subjective.
- Users are not good/bad but just have different opinions.
- Implicitly separate users into groups of similar interests.
- Trust those people most who are in the same group.

TROOTH - Organization

- Users have unique IDs organized as ring.
- Store (item,user,vote)-tuples server-side.
- Calculate user specific trust values and final evaluation client-side.

TROOTH – Voting & Evaluation

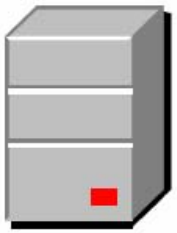
- **Voting**: When a user votes for an item, she sends her opinion (*good* or *bad*) to the TROOTH server and locally adapts the trust values for other users who voted for the same item.

- **Evaluation**: To classify an item, *good* and bad votes from the server are weighted with the client-side stored trust values.

TROOTH - Voting

Item	User	Vote
1	***0***	***good***
1	1	good
1	4	good
1	22	bad
1	83	good
1	114	bad
1	189	good
1	242	good
2	1	bad
…	…	…

Trooth Server



`vote(1,0,bad)`

`(22,bad)`
`(114,bad)`
`(1,good)`
`(4,good)`
`(242,good)`

Trooth Client "0"

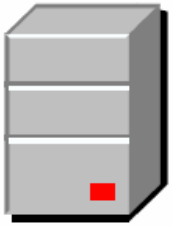
User	Trust
1	***0.15***
4	***0.5***
22	***13***
114	***3.7***
242	***2.2***

User	Trust
1	0.3
22	12.0
114	2.7
242	4.4

TROOTH - Evaluation

Item	User	Vote
2	1	bad
2	4	good
2	22	good
2	83	bad
2	114	good
2	129	good
2	189	bad
2	242	bad

Trooth Server



`eval(2,0)`

Trooth Client "0"

User	Trust
1	0.15
4	0.5
22	13
114	3.7
242	2.2

`(1,bad)`
`(189,bad)`
`(242,bad)`
`(4,good)`
`(22,good)`
`(114,good)`

User	Trust	Vote	
189	1	bad	1 + 2.2 = 3.2 bad
242	2.2	bad	
22	13	good	13 + 3.7 = 16.7 good
114	3.7	good	

=> 16.7 / 19.9 = 0.84 > h_g => **good**

TROOTH - Discussion

- Configurable on client-side
- Number of “known” users is bounded
  - Voting for same type of items
- High burdens for malicious users
  - Voting for same type of items as victim
  - Impact only in direct neighborhood
  - Play by the rules for a long time
- Spamato:
  - SAAS uses challenge/response to assign IDs
  - Votes are signed

Conclusion & Future Work

- TROOTH is a robust, partially decentralized, collaborative, and personalized recommendation system.
- Server-side data could be stored in P2P system

- System is open-source and available for download as part of the Spamato spam filter system: http://www.spamato.net.

Questions?