Trust and Helpfulness in Amazon Reviews

Dylan Shinzaki, Kate Stuckman, Robert Yates

Group 60

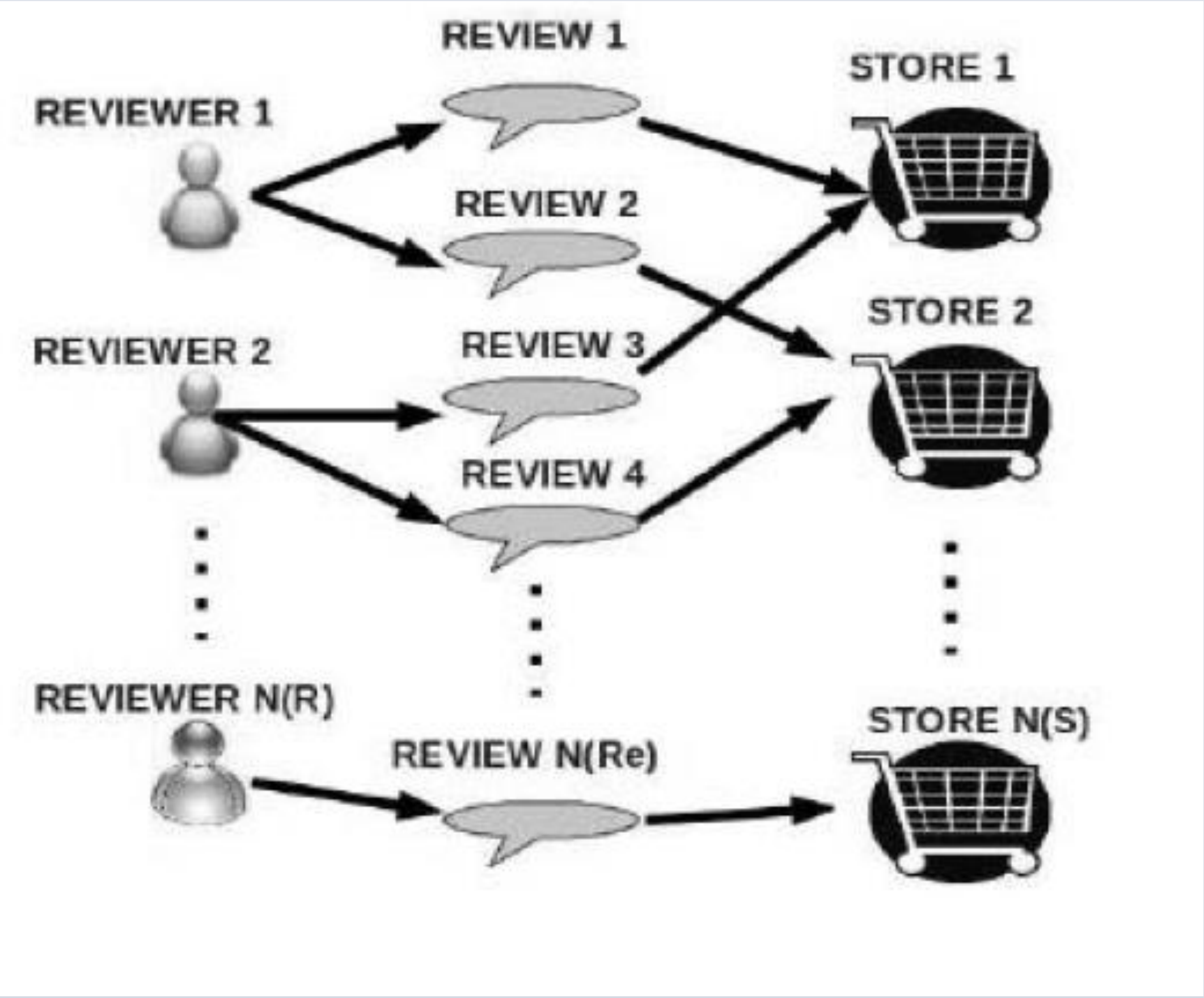
Abstract

- Many Amazon reviews are dishonest spam entries written to skew product ratings
- Users rate reviews as helpful or unhelpful, but this may be influenced by factors other than the review content
- We implement the algorithm proposed in “Review graph based online store review spammer detection” [1] to determine trustworthiness of reviewers
- We create two methods to compute user helpfulness and relate this data to the quantitative assessment of trustworthiness of reviewers
- We improve the efficiency of the algorithm with K approximation
- We evaluate the effectiveness of the algorithm by inserting additional reviewers that model various user behaviors
- All analysis is completed without processing textual content in reviews

Terminology

Term	Description	Symbol	Range
Trustiness	Tendency of a user to supply honest reviews	T	[-1, 1]
Honesty	Tendency of a review agree with other honest reviews	H	[-1, 1]
Reliability	Tendency of a product to get good scores from users with high trustiness	R	[-1, 1]
Agreement	Tendency of a review to match other honest reviews	A	[-1 1]

Review Graph

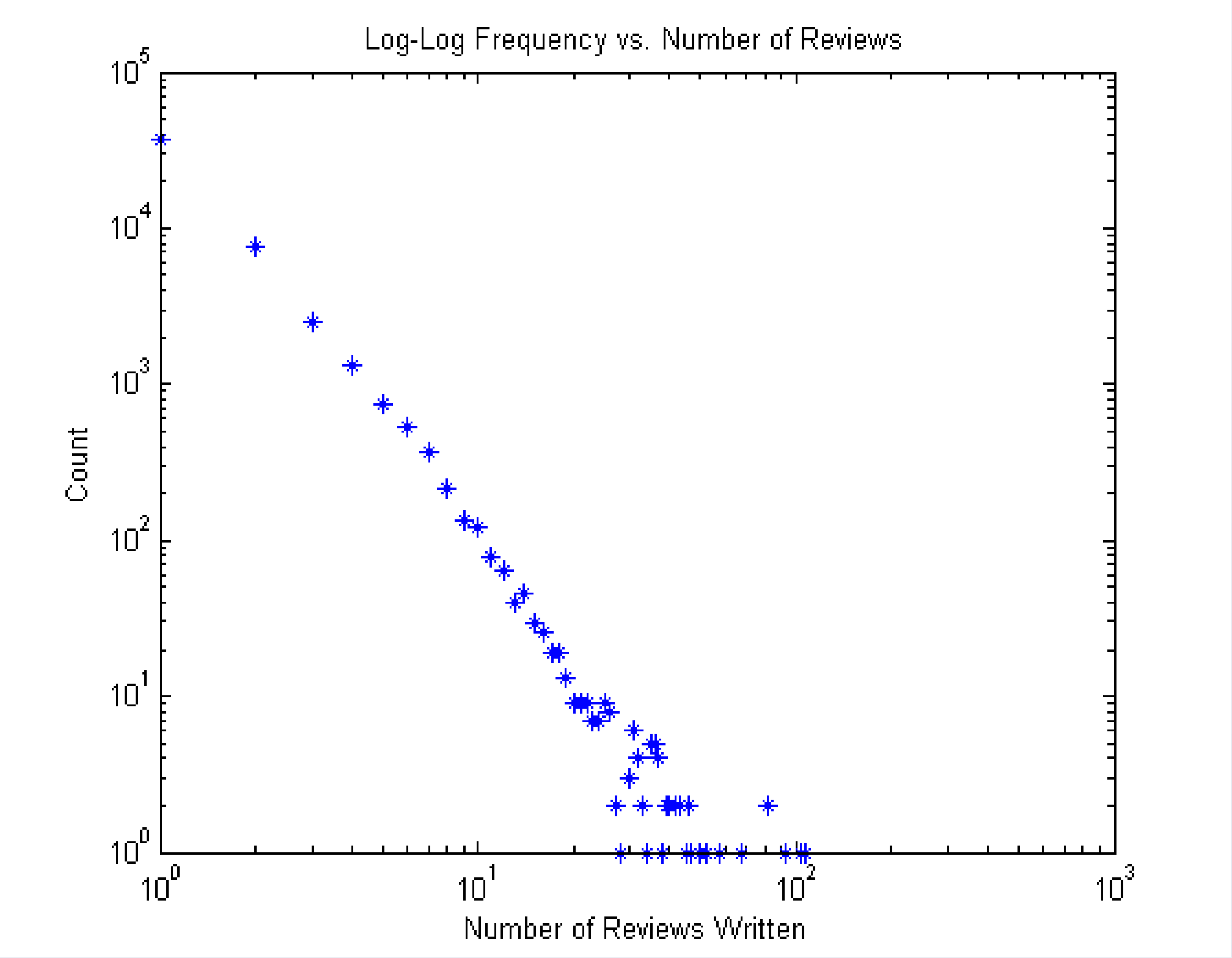


An example of the review graph structure [1]

- The set of reviewers, reviews, and products are interpreted in a graph structure called a “Review Graph”
- There is a node for each reviewer, each review, and each product. The topology of the graph is such that each reviewer node connects with one or more review nodes and each review node connects to exactly one product node.

Dataset

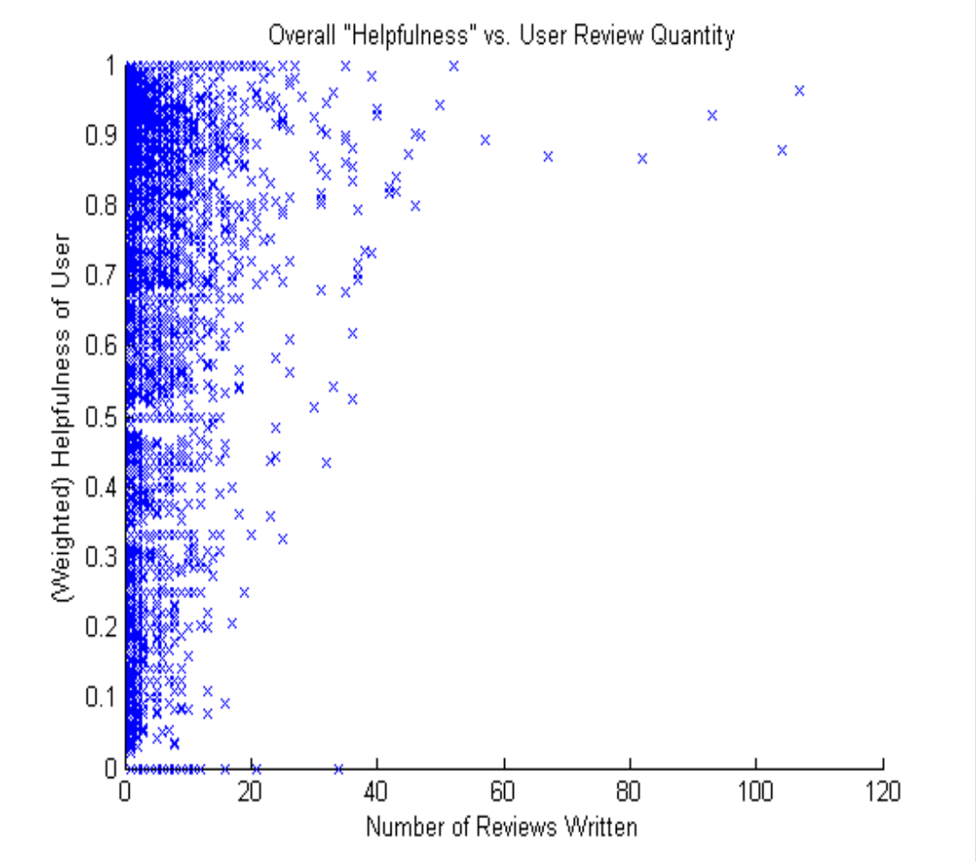
- We studied the Foods subset of an Amazon review dataset
  - The full dataset contains 34,686,770 reviews
  - The Foods subset contains over 500,000 reviews
  - The number of reviewers per user follows a power law distribution



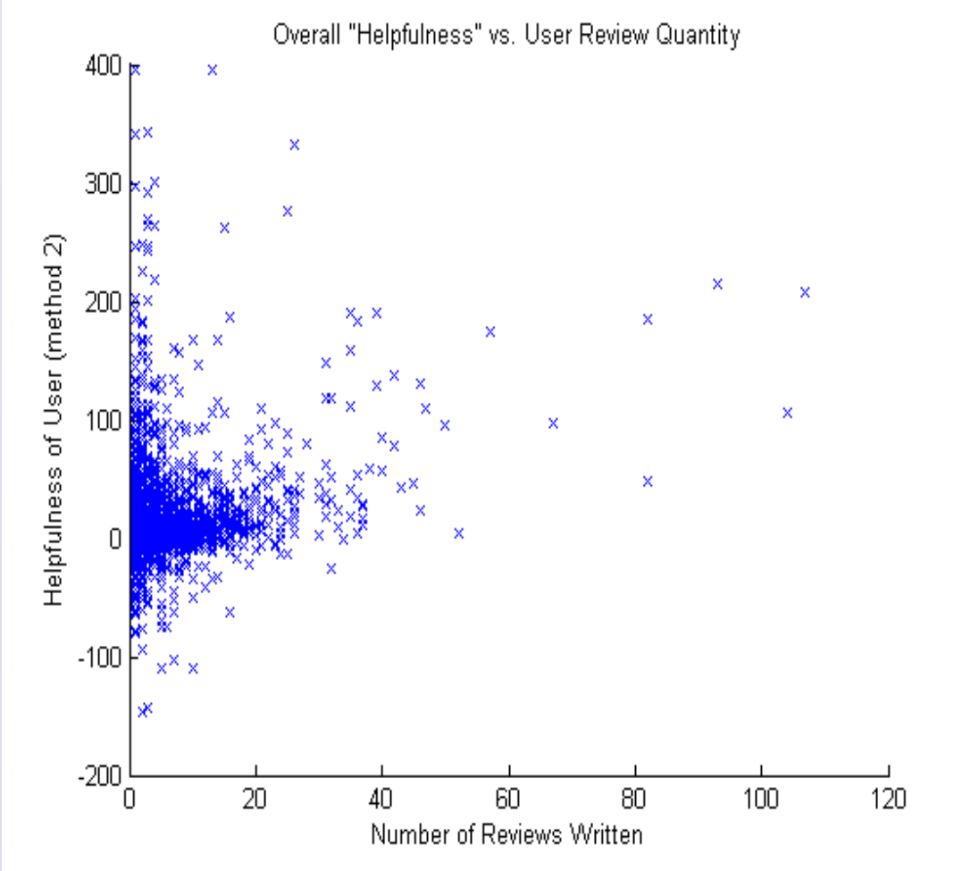
Computing User Helpfulness

Method 1 of computing helpfulness involves taking the sum of positive helpfulness votes over the total sum of helpfulness votes.

Method 2 of computing user helpfulness involved taking the sum of the positive helpfulness votes minus the sum of the unhelpful votes over all of a user’s reviews.



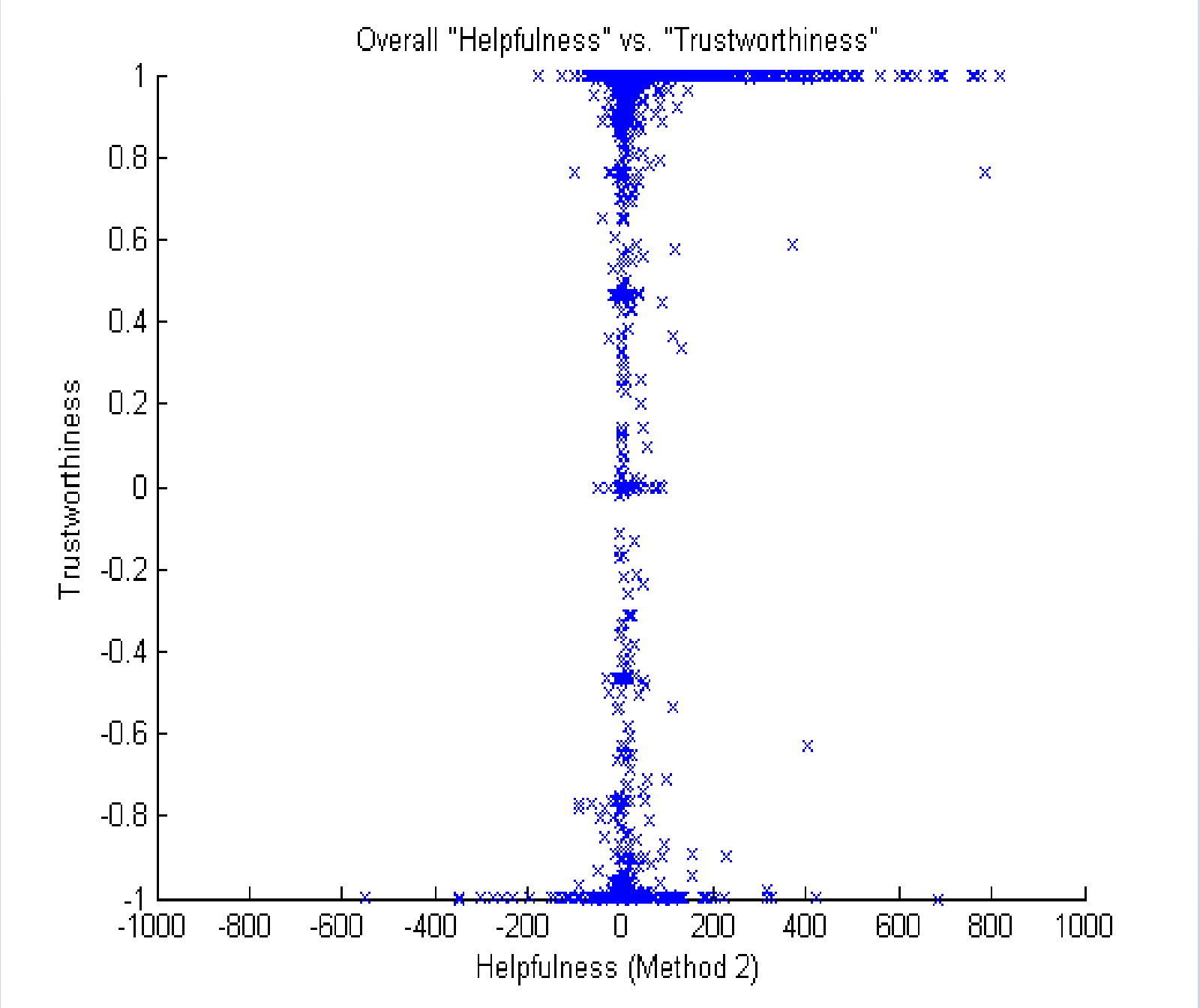
Method 1



Method 2

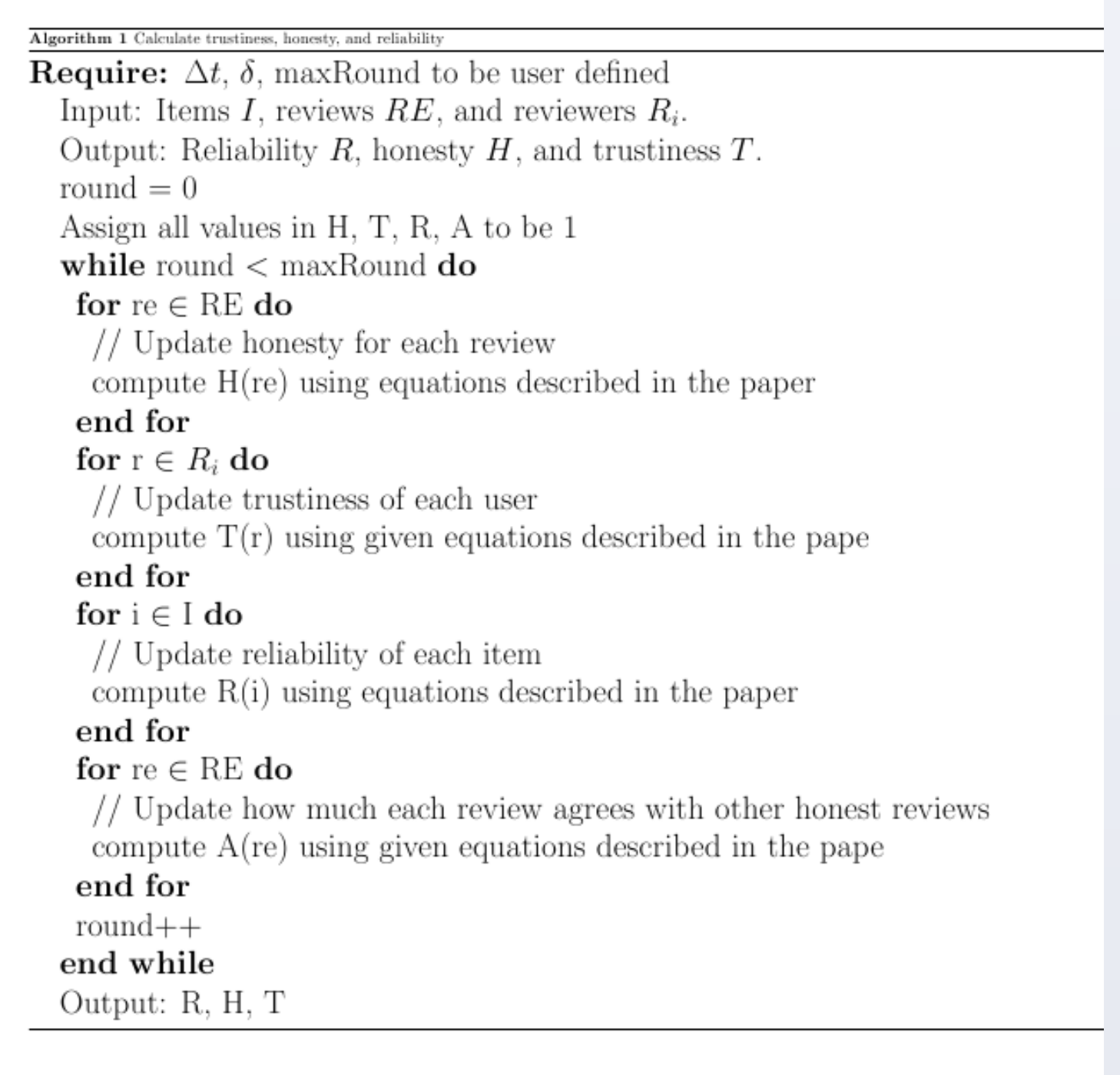
Relating Helpfulness and Trust

Correlation is strongest Method 2 when only outlier values are considered. For example, 87% of users with helpfulness less than -100 have trustworthiness ratings less than -0.9. This could prove useful in spam detection.



Algorithm

- The trustiness, reliability and honesty of each user, product and review is calculated though iterative update. Each value is initially set to 1 and iteratively updated using a set of equations which interdependently relate these values.



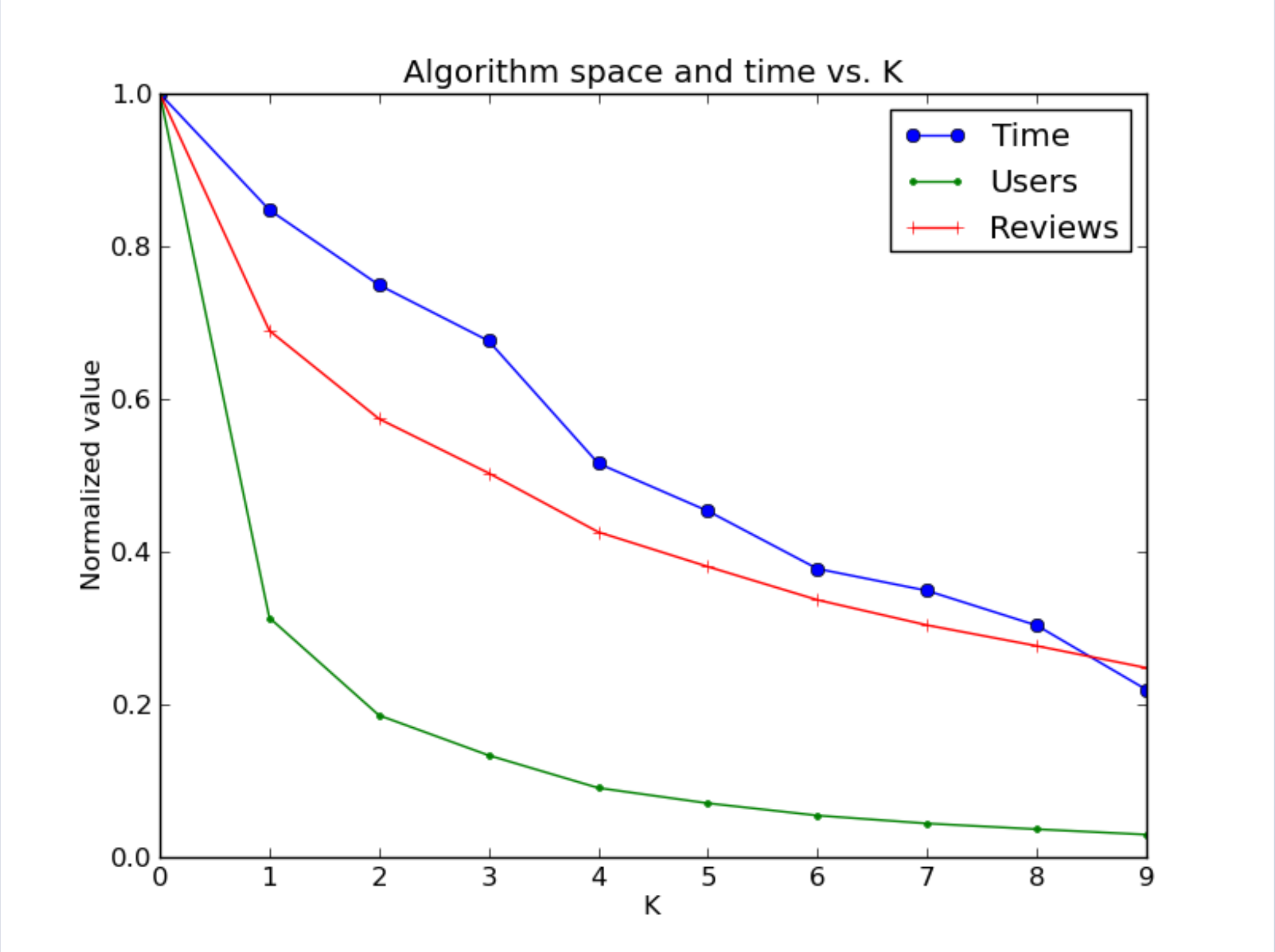
```
Algorithm 1 Calculate trustiness, honesty, and reliability
Require: $\Delta t$, $\delta$, maxRound to be user defined
  Input: Items $I$, reviews $RE$, and reviewers $R_i$.
  Output: Reliability $R$, honesty $H$, and trustiness $T$.
  round = 0
  Assign all values in H, T, R, A to be 1
  while round < maxRound do
    for re ∈ RE do
      // Update honesty for each review
      compute H(re) using equations described in the paper
    end for
    for r ∈ $R_i$ do
      // Update trustiness of each user
      compute T(r) using given equations described in the pape
    end for
    for i ∈ I do
      // Update reliability of each item
      compute R(i) using equations described in the paper
    end for
    for re ∈ RE do
      // Update how much each review agrees with other honest reviews
      compute A(re) using given equations described in the pape
    end for
    round++
  end while
  Output: R, H, T
```

- An example is the calculation of trustiness of a given user where Hr is the sum of the honesty scores of over all of the user's reviews

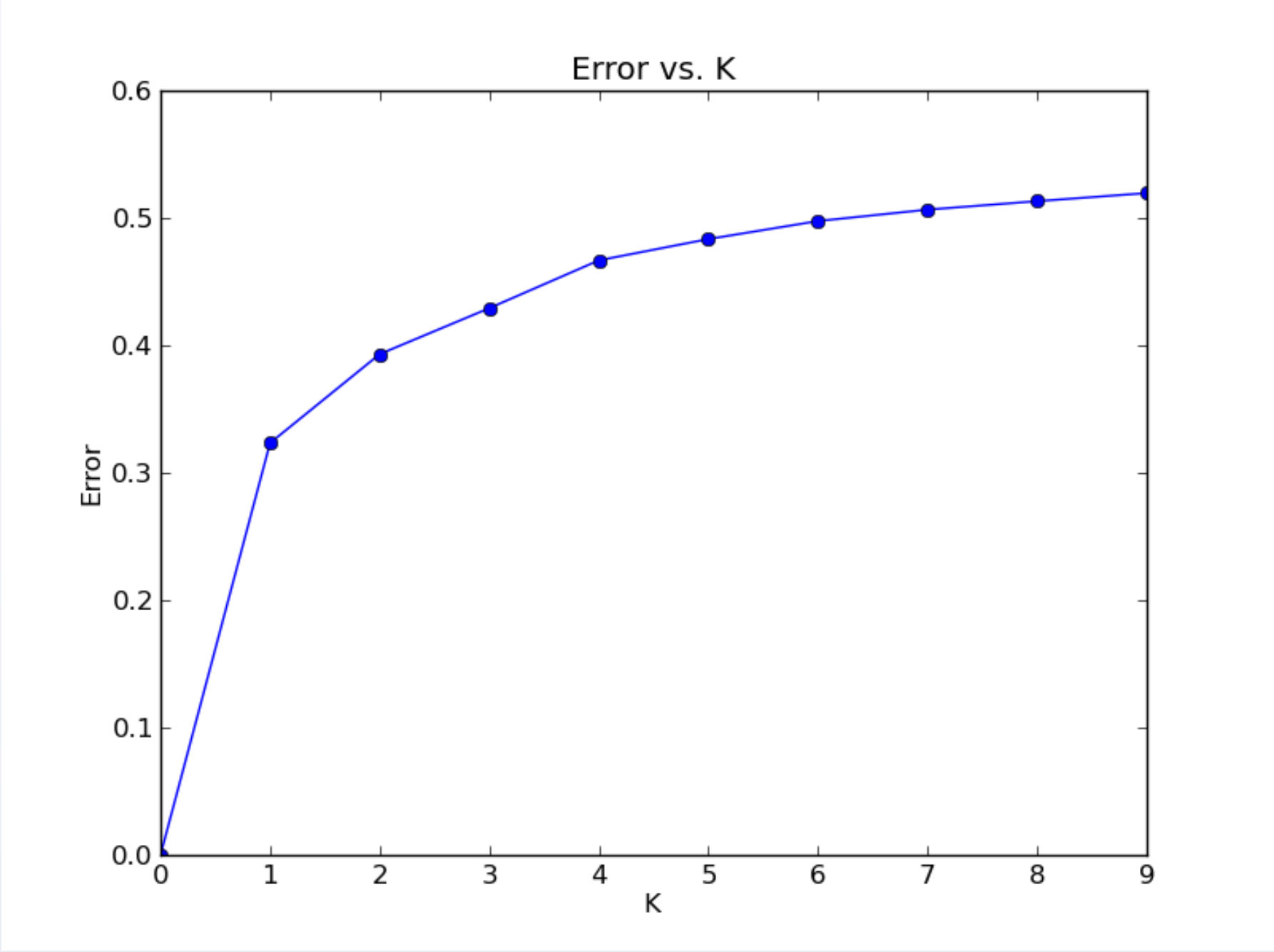
$$T(r) = \frac{2}{1+e^{-H_r}} - 1$$

K-Approximation

- Size is a major consideration when doing calculations on this type of graph.
- This corresponds to ignoring users who have written less than k reviews by always assigning them a trust of 0
- Can yield a substantial decrease in practice, though there are no theoretical guarantees as the savings depend heavily on the graph topology.



Normalized problem size vs. K



Error Vs. K

Evaluation

- To evaluate the algorithm, the dataset was seeded with researcher generated users who generated reviews from the following models. Such users had the following trustiness values

Model	Description
Downvote Bot	Always review 1.0
Upvote Bot	Always reviews 5.0
Conformist	Always reviews the average score
Random	Score taken uniformly at random from 1.0 to 5.0 inclusive

Model	Average	Standard deviation	Median
Downvote Bot	-0.768	0.381	-0.094
Upvote Bot	0.941	0.096	0.073
Conformist	0.946	0.053	0.942
Random	-0.334	0.603	-0.613

User models and their performances

- While the Downvote bot and Conformist had expected behavior, the Upvote bot did unexpectedly well. The Random model was not 0.
- This can be explained by the score breakdown of dataset. The majority of scores are 5.0, meaning that the Upvote bot often appears to be agreeing with the majority.
- This shows a weakness in a reputation-only trust evaluation. This highlights the need for other approaches, like those that analyze review text for key spam phrases.

Score	1	2	3	4	5
Count	51691	29430	42090	79428	357730
Percentage	9.22%	5.25%	7.51%	14.20%	63.80%

Score breakdown of the Fine Foods dataset

Selected References

[1] G. Wang, S. Xie, B. Liu, and P. S. Yu, “Review graph based online store review spammer detection," in Data Mining (ICDM), 2011 IEEE 11th International Conference on, pp. 1242- 1247, IEEE, 2011.

Acknowledgements

We’d like to acknowledge Professor Leskovec for teaching this class. Additionally, we’d like to thank Ashley Jin for her helpful comments and recommendations.