Modeling Agent Trustworthiness with Credibility for Message Recommendation in Social Networks[*]

(Extended Abstract)

Noel Sardana
School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
nisardan@uwaterloo.ca

Robin Cohen
School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
rcohen@uwaterloo.ca

ABSTRACT

This paper presents a framework for multiagent systems trust modeling that reasons about both user credibility and user similarity. Through simulation, we are able to show that our approach works well in social networking environments by presenting messages to users with high predicted benefit.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

Keywords

Trust Modeling, Credibility, Recommending Messages

1. BACKGROUND

In this paper, we choose as our primary competitor the trust model of Champaign et al. [1] called LOAR (Learning Object Annotation Recommendation). It is inspired by Zhang's personalized approach to trust modeling [2] which integrates both private and public reputation of peers but is designed to operate well in social networking environments in order to judge the trustworthiness of annotations left on web objects by incorporating additional elements, namely: (i) a modeling of peer similarity based on past rating behaviour (where peers view messages and then downvote or upvote) (ii) a modeling of an annotation's reputation (based on one of three combination functions to compute the reputation of the annotator (based on past reaction to his annotations, where votes are scaled up or down based on peer similarity).

2. MODEL

Our central algorithm for message recommendation is displayed in Algorithm 1. Determining whether an annotation or message will be well-received (i.e., is beneficial) is not deterministic; it can instead be modelled as a Bernoulli process. That is, if M is the event that a message is well-received, then we seek to determine $\psi = Pr(M)$. Moreover, we allow this parameter to itself be represented as a random variable and rely on Bayes' theorem to update prior probability distributions over ψ. In particular, we can use the beta distribution[1] to represent the prior $Pr(\psi)$, so that $\psi \sim Beta(\alpha^*, \beta^*)$. However, since we model the trustworthiness of messages (not annotators), the user does not have any prior belief that directly corresponds to the message itself (has yet to experience it, so the only rational belief is to assume that $\alpha^* = \beta^* = 1$, i.e., that ψ is uniformly distributed in the interval $[0, 1]$). When a user solicits feedback about a message, his peers report binary ratings. Equivalently, peers report parameters α_p and β_p such that $\alpha_p + \beta_p = 1$. In this work, we restrict this report such that $\alpha_p, \beta_p \in \{0, 1\}$. To combine peer reports, we model the similarity between users i and j using Hamming distance. The Hamming distance is a measure of the number of bits by which two binary strings differ, or equivalently, how many changes need to be made to string a to transform it into string b. Here, we can consider the series of common annotation ratings between two users to form "binary rating strings".

We normalize the Hamming distance between i and j to arrive at a similarity metric called the Hamming ratio, denoted h_{ij} (the Hamming distance divided by the length of the binary strings, i.e., the number of common ratings). Since a Hamming distance of 0 means that the two strings are identical, a Hamming ratio of 0 suggests we simply take a peer report as given; in contrast, if the Hamming ratio is 1, we swap the values reported for α_p and β_p. This captures the fact that non-similar peers can still deliver useful information; perfect negative correlations are just as informative as positive ones. We formalize this combination as follows:

$$\alpha^* = 1 + \sum_{p \in P} (1 - h_{sp}) \cdot \alpha_p + h_{sp} \cdot \beta_p \tag{1}$$

$$\beta^* = 1 + \sum_{p \in P} (1 - h_{sp}) \cdot \beta_p + h_{sp} \cdot \alpha_p \tag{2}$$

A report $r \in [0, 1]$ can be translated into parameters $(\alpha, \beta) = (r, 1 - r)$ so that a report of $r = 1$ corresponds to $\alpha =$

[*] Financial assistance was received from NSERC.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

[1] In particular, α^* (β^*) represents the strength of a user's belief that a message will be good (bad). Thus, when α^* is high and β^* is low (β^* is high and α^* is low), the user will be very confident that he should see (not see) the message.

$1, \beta = 0$, a report of $r = 0.5$ corresponds to $\alpha = \beta = 0.5$, and $r = 0$ to $\alpha = 0, \beta = 1$. To make this more explicit, suppose that user i solicits advice from user j about a message m. Suppose further that the Hamming ratio between i and j is 1 (that is, they are completely opposite). Then, if j reports $(\alpha, \beta) = (0, 1)$ (i.e., he thinks the message not useful, or perhaps even incorrect), the similarity weighting scheme described above will reverse this opinion to $(\alpha, \beta) = (1, 0)$ when determining the trust metric from i's perspective. That is, the message, which j thinks should not be shown, will now be more likely to be shown. However, in this case, if j is perfectly credible, his opinion of a message corresponds to a very credible one. Accordingly, his report might be better taken verbatim rather than dampened by the Hamming ratio. This algorithm encodes a heuristic when amalgamating peer advice. We typically heed peer advice to the extent that the advisors have similar preferences to our own. However, CredTrust reverses the role that similarity plays in overturning our interpretation of peer advice, so that non-similar but credible peer advice will be heeded, verbatim. Thus, credible peers can stop folklore: propagation of false messages by similar (non-credible) peers.

Algorithm 1: Deriving a predicted benefit using similarity and credibility (CredTrust)

```
Input: The current user, $u$, his set of peers, $P$, their
       credibility scores, $c_p \in [0, 1]$, and their
       corresponding ratings for the annotation in
       focus, $r_p \in \{0, 1\}$
Output: Parameters $\alpha^*$ and $\beta^*$ to a beta distribution
        describing trust in the current annotation
$\alpha^* = \beta^* = 1$ // At the start, user has a uniform
    expectation about the message
foreach $p \in P$ do
    $h_{up} \longleftarrow computeHammingRatio(u, p)$
    // Perform a Bayesian update after
        discounting heuristic
    if $r_p == 0$ then
        // Adjust the similarity weight by
            credibility:
        $\alpha^* += h_{up}(1 - c_p)$
        $\beta^* += 1 - h_{up} \cdot (1 - c_p)$
    else
        // Dampen update by credibility
        $\alpha^* += c_p \cdot (1 - h_{up})$
        $\beta^* += c_p \cdot h_{up}$
    end
end
```

3. VALIDATION

We simulate an environment consisting of 20 agents, each of whom create messages and rate messages created by other the agents. Agents are partitioned into one of two sets: low credibility or high credibility. When authoring messages, credibility scores influence the "underlying message credibility" of the messages the agents create. For example, when an agent has a credibility of 0.5, approximately half of the messages it authors will be simulated to be beneficial and approximately half of the messages will have a "flaw" that detracts from agents' utilities if read.

In addition to credibility, agents are randomly assigned a type $\theta_a \in [0, 1]$. The agent's type is a parameter that influences similarity; agents of the same type tend to like the same messages. Moreover, messages have a type $\theta_m \in [0, 1]$ in order to appeal to different agents. In particular, we simulate agents rating messages more highly when those messages correspond to their type. However, agents' evaluation of the credibility of each message is modeled by flipping biased coins with probabilities proportional to their own credibilities; if an agent considers a message to be credible, and that message closely matches the agent's type, it will rate the message highly. The result is that less credible agents tend to rate messages they like highly, irregardless of any misinformation or flaws contained within the message.

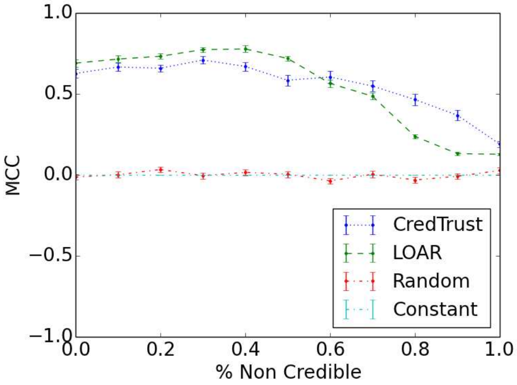

Figure 1: MCC versus percentage of non-credible advisors.

Each agent randomly produces between 1 and 10 messages and rates all of the messages produced by other agents. In order to evaluate the quality of the inferred benefits for messages, we randomly partition messages into a training and validation set. The training set is composed of approximately 70% of the messages and is used for the purpose of determining the Hamming distances (for CredTrust) and the similarities and author reputations (for LOAR).

Once all of the algorithm inputs have been computed, the simulation runs each algorithm on the testing set to find the predicted benefits for each message based on the advisory ratings. If the predicted benefit of a message is determined to be high (i.e., greater than 0.5), the message is recommended; otherwise, it is rejected. We compute the number of correctly classified messages (i.e., correctly recommended or correctly rejected) by comparing to the "correct" message classifications (based on the known benefits of each message to each agent) and report the Matthew's Correlation Coefficient (MCC), which relates the true positive, false positive, false negative, and true negative rates. Results show that CredTrust performs well and outperforms LOAR.

4. REFERENCES

[1] J. Champaign, J. Zhang, and R. Cohen. Coping with poor advice from peers in peer-based intelligent tutoring: The case of avoiding bad annotations of learning objects. In *Proceedings of User Modeling, Adaptation and Personalization*, pages 38–49, 2011.

[2] J. Zhang and R. Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, pages 330–340, 2008.