SONM User Reputation Model

Andrei Zavgorodnii, Anastasia Ashaeva

Jan, 2017

Contents

1 Overview

This document describes the reputation model that is used to characterise agents in the SONM network. Motivation: lost profit for sellers, lost time for buyers, etc.

1.1 Basic notions

The SONM ecosystem consists of *buyers* and *sellers*. Buyers rent computational resources from sellers to run arbitrary *tasks*; a deal is made for a specific resource configuration and a specific period of time (i.e., not per task).

When looking for a seller, buyer searches the *marketplace*. When a deal is made, a certain amount of funds is reserved on both buyer's and seller's accounts. For buyer, it's the *full price* of the deal; for seller, it's a fraction of the full price (either default or negotiated), hereinafter *the deposit*.

Both buyers and sellers have *reputation* that is based on their activity in the SONM network, i.e., on the outcomes of public deals they had. Any public deal can have three possible outcomes:

- **Mutual satisfaction.** Buyer is satisfied with the service provided by seller. Buyer's and seller's reputation increase in proportion to deal price.
- **Settled dispute.** Buyer is *not* satisfied with the service provided by seller, seller admits his fault. Buyer pays seller in proportion to the time it held seller's resources, and seller returns buyer the deposit.
- **Claim.** Buyer is *not* satisfied with the service provided by seller, seller refuses to admit his fault. Both buyer and seller keep their money, but lose their ratings in proportion to deal's price.

Both buyers and sellers are motivated to have high reputation. Firstly, reputation (combined with deals history) defines the maximum price of a deal you can make (see **??**). Secondly, reputation defines the order in which offers are displayed on the market, so agents with higher reputation will have better deals.

Deals can be *public* and *private*. Private deals are made directly between buyers and sellers (i.e., bypassing the marketplace) and, importantly, *do not affect user's reputation.* See section **??** for cases when private deals come into play.

2 Reputation

2.1 The model

SONM's reputation model is based on beta distribution's expected value, following the ideas proposed in [**?**]. The beta-family of probability density functions is a continuous family of functions indexed by the two parameters α and β. The beta distribution $f(p|\alpha,\beta)$ can be expressed using the gamma function Γ as:

$$\phi(p|\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1},\ \text{where } 0\leq p\leq 1,\ \alpha>0,\ \beta>0. \tag{1}$$

The probability expectation value of the beta distribution is given by:

$$E[p]=\frac{\alpha}{\alpha+\beta}. \tag{2}$$

Suppose that we take n and m to be the number of positive and negative outcomes for user X respectively; then we define α and β:

$$\alpha=n_X+1,\ \beta=m_X+1. \tag{3}$$

Note that in (**??**) we start counting from 1 for both successful and unsuccessful outcomes to set initial reputation to 0.5.

Positive outcomes correspond to successful deals, and negative outcomes correspond to claims. Then reputation for user X is defined as probability expectation value of (**??**):

$$\text{Rep}_X=E[\phi(p|\alpha,\beta)]=\frac{n+1}{n+m+2}. \tag{4}$$

To control how fast reputation grows at the start of deals history, so we introduce the *weight* coefficient $\gamma\in(0,1]$ which modifies the impact of successful deals on reputation n (see [**?**] for how γ affects reputation curvature):

$$\text{Rep}_X=E[\phi(p|\alpha,\beta)]=\frac{\gamma n+1}{\gamma n+m+2}. \tag{5}$$

In (**??**) we assume that n and m are calculated by simply counting positive and negative outcomes, but it's natural to take deal price into consideration. Suppose that ξ_X is the history of X's deals, and ξ_{Xi} is the price of i-th deal (negative signed if the deal ended in conflict). Let's define helper functions to tell successful deals from unsuccessful:

$$\omega^{+}(\xi, i) = \begin{cases} \xi_{Xi}, & \text{if } \xi_{Xi} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

$$\omega^{-}(\xi, i) = \begin{cases} |\xi_{Xi}|, & \text{if } \xi_{Xi} \ 0, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Then deal outcomes can be weighted by their price as:

$$n_X = \sum_{i=1}^{n} \omega^{+}(\xi_X, i) \tag{8}$$

$$m_X = \sum_{i=1}^{n} \omega^{-}(\xi_X, i), \tag{9}$$

where n is the total number of deals made by X. Now that we count weights (and not just outcomes) we must update (**??**) to stabilize reputation growth in the very beginning of deals history; instead of adding one history unit we add user's *mean deal price*[1]:

$$\alpha = n_X + \mu_{\xi_X},\ \beta = m_X + \mu_{\xi_X}. \tag{10}$$

We might want older outcomes to contribute less to user's reputation. We introduce another parameter, λ, as the *forgetting factor*:

$$n_{\lambda}^{X} = \sum_{i=1}^{n} \omega^{+}(\xi_X, i) \cdot \lambda^{(n-i)} \tag{11}$$

$$m_{\lambda}^{X} = \sum_{i=1}^{n} \omega^{-}(\xi_X, i) \cdot \lambda^{(n-i)}, \tag{12}$$

where $0 \leq \lambda \leq 1$. It's easy to see that if $\lambda = 1$, nothing is forgotten.

So, the final version of reputation function looks as follows[2]:

$$\mathrm{Rep}_X = \frac{\gamma[\sum_{i=1}^{n} \omega^{+}(\xi_X, i) \cdot \lambda^{(n-i)}] + \mu_{\xi_X}}{\gamma[\sum_{i=1}^{n} \omega^{+}(\xi_X, i) \cdot \lambda^{(n-i)}] + \sum_{i=1}^{n} \omega^{-}(\xi_X, i) \cdot \lambda^{(n-i)} + 2 * \mu_{\xi_X}}. \tag{13}$$

[1] We set reputation to zero if there's no deals history.

[2] See how this definition behaves in different scenarios in **??**.

2.2 Reputation an fraud rate

Fraud rate is a rational number from $(0, +\infty)$ that is proportional to the probability of having an unsuccessful deal of price p with user X:

$$F(p) = \frac{p}{\mathrm{Rep}_X \cdot \sum_{i=1}^{n} \xi_{Xi}} \tag{14}$$

3 Threat Model

It is well known [?, ?] that reputation systems are subject to collusion and fraud, or at least that such threat always exists. Since a higher reputation means more profit, some users try to fraudulently increase their reputation; alternatively, a user might pick the strategy to eventually trade some of his reputation for immediate profit. This section describes various cases that fall into either of these scenarios, and the proposed means to counter such behaviour.

3.1 Fraudulent buyers

A malicious buyer can open a claim after the seller has fulfilled his obligations in good faith, which will result in reputation loss for both buyer and seller.[1] There's two different cases to consider: when there's a loner fraudulent user and when there's a group of users who behave fraudulently together.

3.1.1 Single buyer

For an isolated fraudulent buyer the motivation is to get resources for free by losing some of his reputation. It is important that to gain any significant profit from such actions (i.e., to make a deal of a significant price and refuse to pay for it) a buyer has to have a relatively long history of successful deals (see **??**).

Now, from the seller's perspective, dealing with a fraudulent buyer means lost profit and reputational loss, but there's a difference when it comes to reputation: an average seller has much more deals in his history than any client; this means that in case of a claim a seller's reputation will be affected to a much lesser extent than that of a buyer. The buyer's reputation, on the other hand, will go down significantly, and it will take significant time to recover (see **??** for detailed illustrations). Moreover, a seller can add a user to *blacklist* and refuse to deal with him/her in the future; blacklisting will make it hard for a buyer to consistently fall back to fraudulent behaviour.

3.1.2 A group of buyers

As we could see from the previous section, single fraudulent buyer can not affect a seller's reputation significantly. There is, however, a scenario when a group of users work together to reduce some

[1] We assume that a diligent seller won't be willing to admit a non-existent fault and lose his deposit.

seller's reputation. Such group is most likely to consist of "bots", i.e. users that were created specifically for fraudulent purposes.[2]

For example, if for a certain market segment the supply is greater than demand, a seller might want to spoil his competitor's ratings to get better deals. Then a botnet can be launched to open deals with a certain seller and then refusing to accept the results. However, there's several factors that make this difficult to achieve.

Firstly, SONM uses a "know your client" system to authenticate buyers. One of the easiest ways to pass the KYC check is to bind your account to a bank card; this means that to create a substantial botnet one will need to have access to many cards, which is obviously troublesome (if not illegal).

Secondly, if a user does manage to create lots of authenticated bots, he will have to gain enough reputation for each of those bots. To avoid wasting a lot of money, a seller will just make deals with himself. There is, however, a simple way to cope with such fraud: for any buyer, only N *public* deals are allowed to be made with the same seller within a month. Subsequent deals have to be private and will not affect buyer's rating.

3.2 Fraudulent sellers

A fraudulent seller might provide incomplete service to a customer and gain some profit. This is possib

[2] Apart from the case when users declare war on a seller for some external reason, which we are not going to discuss here.

4 Appendix

4.1 Basic reputation scenarios

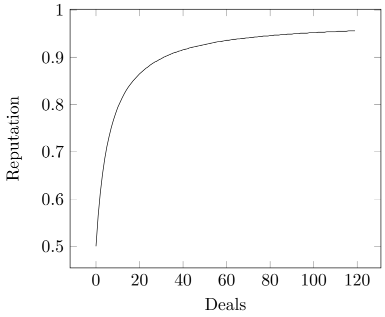


Fig. 1 (uniform, stable)

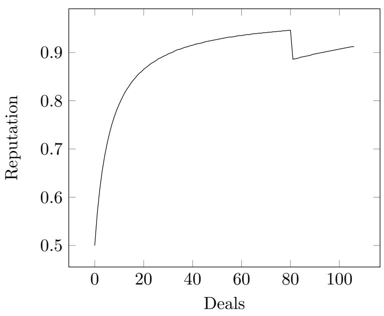


Fig. 2 (uniform, 1 conflict in a row)

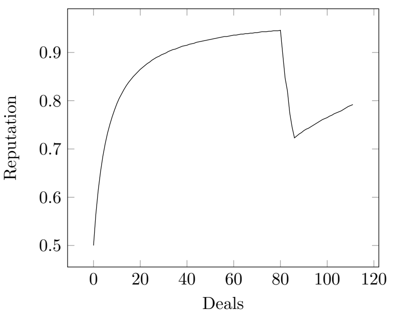


Fig. 3 (uniform, 6 conflicts in a row)

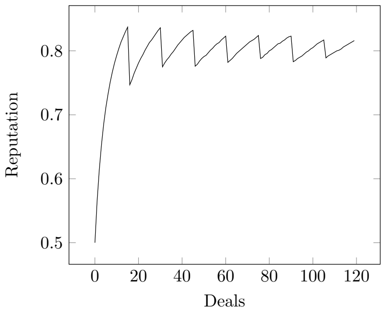


Fig. 4 (uniform, each 15-th conflict)

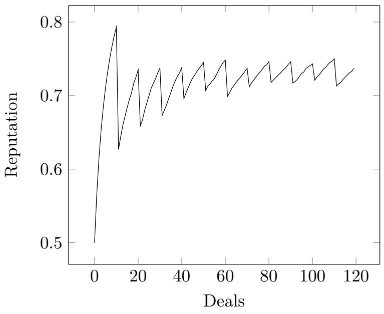


Fig. 5 (uniform, each 5-th conflict)

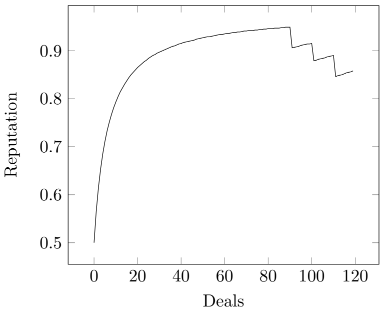


Fig. 6 (uniform, decay to each 10-th conflict)

Fig. 1

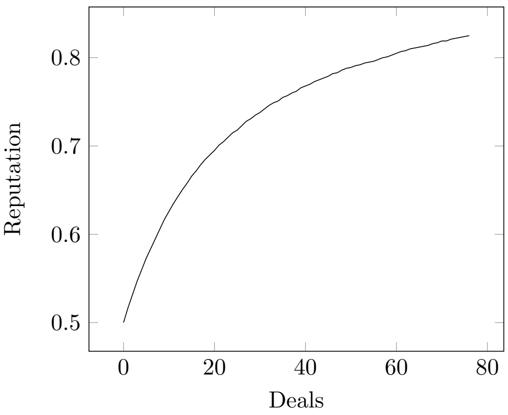


Fig. 1 (Mean price = 91, Turnover = 7051)

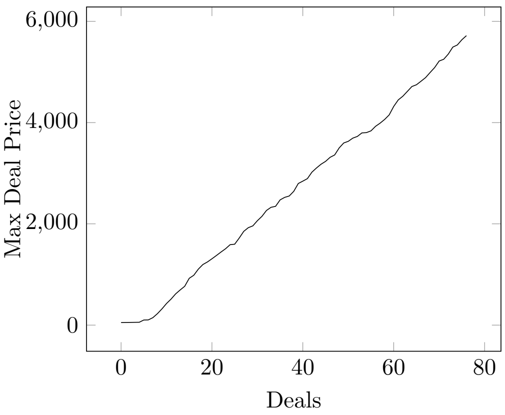


Figure 1 shows a simplified scenario of how a user's reputation grows if he makes evenly priced deals. Figure 2 shows how a user's maximum deal price grows as he makes successful deals.

4.2 Single buyer: fraud and recovery scenarios

Fig. 1 (uniform,decay to each 3-th conflict)

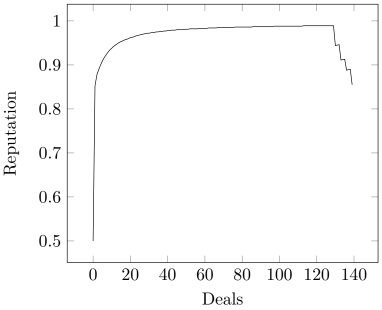


Fig. 2 (uniform, decay to each 10 conflict)

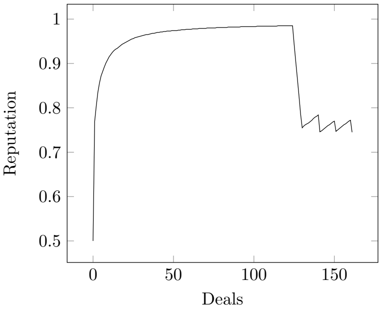


Fig. 3 (uniform, decay to each 5-th conflict)

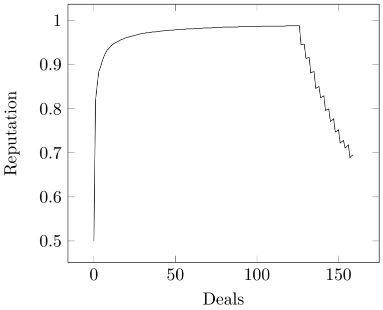


Fig. 4 (uniform, stable)

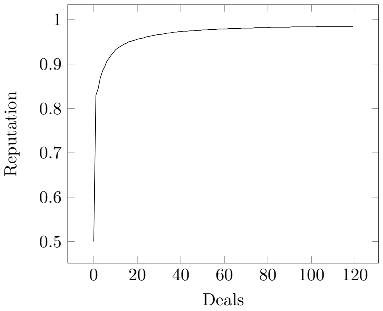


Fig. 5 (uniform, 6 conflicts in a row)

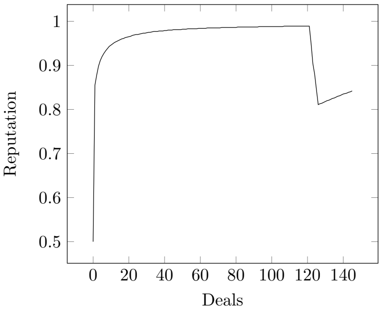
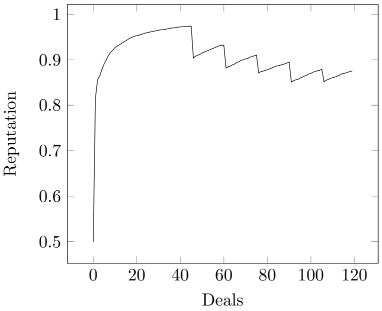


Fig. 6 (uniform, each 15 conflicts in a row)



Lorem ipsum.

References

[1] Josang, Audun and Ismail, Roslan. *The beta reputation system.* Proceedings of the 15th bled electronic commerce conference, vol. 5, p. 2502–2511, 2002.

[2] Ciccarelli, Gianluca and Cigno, Renato Lo. *Collusion in peer-to-peer systems.* Computer Networks, vol. 55, p. 3517–3532, 2011.

[3] Maranzato, Rafael and Pereira, Adriano and do Lago, Alair Pereira and Neubert, Marden. *Fraud detection in reputation systems in e-markets using logistic regression.* Proceedings of the 2010 ACM Symposium on Applied Computing, p.1454–1455, 2010.