

# Using Jiminy for Run-Time User Classification Based on Rating Behaviour

Evangelos Kotsovinos<sup>1</sup>, Petros Zerfos<sup>1</sup>, Nischal M. Piratla<sup>1</sup>, and  
Niall Cameron<sup>2</sup>

<sup>1</sup> Deutsche Telekom Laboratories  
`firstname.lastname@telekom.de`

<sup>2</sup> Pembroke College, Cambridge  
`niall.cameron@cantab.net`

## 1 Introduction

This paper describes an application of our prototype implementation of *Jiminy*, a scalable distributed architecture for providing participation incentives in on-line rating schemes. Jiminy is based on an incentive model where participants are explicitly *rewarded* for submitting ratings, and are *debited* when they query a participating reputation management system (RMS). Providing explicit incentives increases the quantity of ratings submitted and reduces their bias by removing implicit or hidden rewards, such as those gained through revenge or reciprocal ratings. To prevent participants from submitting arbitrary or dishonest feedback for the purpose of accumulating rewards, Jiminy halts rewards for participants who are deemed dishonest by its probabilistic *honesty estimator*. Using this estimator, Jiminy can also perform *classification* of users based on their rating behaviour, which can be further used as criteria for filtering the rating information that users obtain from the RMS.

More background on the theoretical foundations of Jiminy can be found in [1], while [2] provides details on the system design, implementation and performance evaluation.

## 2 Application Scenario

Jiminy follows a cluster-based architecture and is deployed on a number of computers, for real-time computation of the statistical analysis of ratings. This allows online monitoring and classification of the rating behaviour of users. Jiminy operates in the following steps:

**Bootstrapping.** Jiminy starts its operation by connecting to the ratings database, discovering the available slaves, and deciding how to partition the problem space — we use the Grouplens<sup>1</sup> ratings data set — among the slaves, based on the number of slaves available. It then communicates to each slave the part of the problem space it is assigned, and starts a network daemon listening to requests from the reputation management system (RMS).

---

<sup>1</sup> <http://www.grouplens.org>

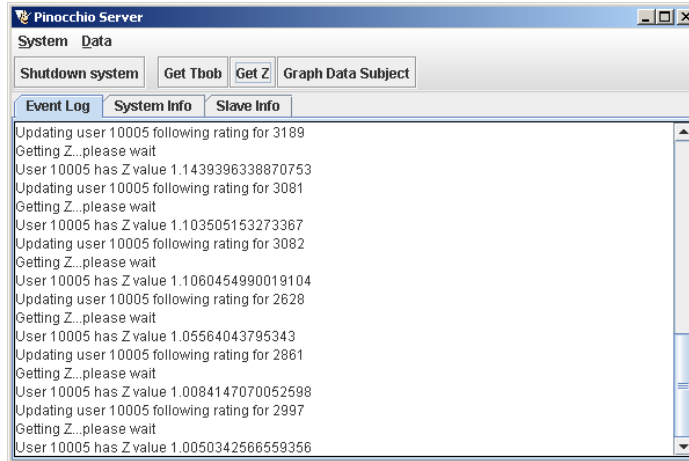


Fig. 1. Interface of the Jiminy master during periodic Noselength updates

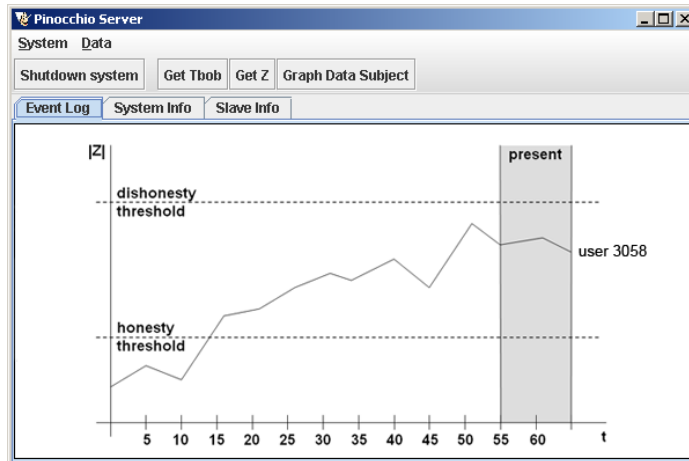
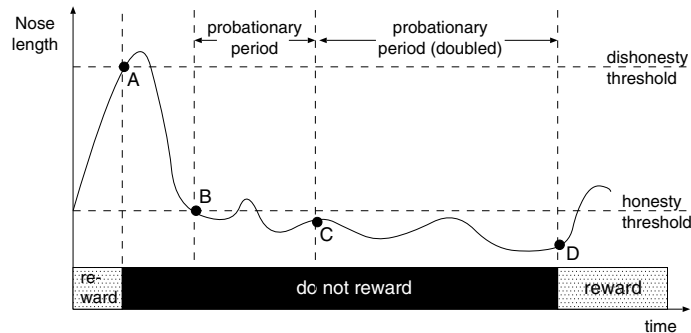


Fig. 2. Run-time monitoring of a user’s Noselength, and classifications

**Periodic Honesty Recalculation.** Jiminy periodically runs the algorithm that calculates and updates the *Noselength* value for each user — our honesty metric, as described in [2]. The Jiminy GUI provides a log of honesty recalculation events — as shown in Figure 1. It also displays a certain user’s Noselength value by producing a graph of his Noselength in real time, as shown in Figure 2. Submitting dishonest ratings will have a negative effect on the Noselength of the user who submitted them, which can be reversed by submitting honest ratings.



**Fig. 3.** Honesty and dishonesty thresholds, and probationary period

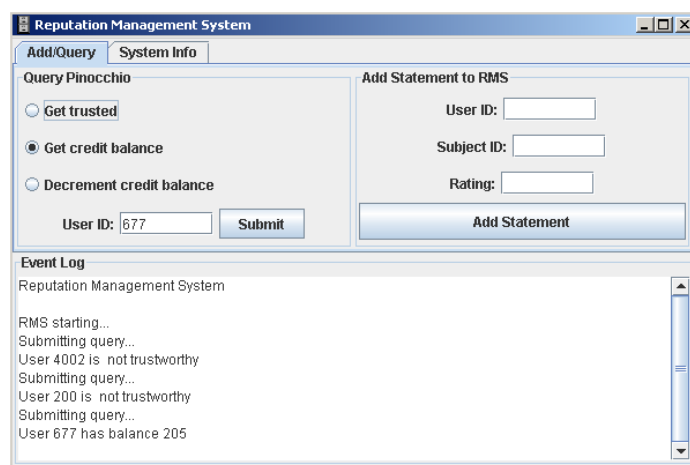
**Probation and Classification.** At the end of each honesty recalculation iteration, Jiminy identifies whether each user is to be considered honest or dishonest, and whether she is to face a probationary no-rewards period, as follows:

- A user stays outside the probationary period, if she is currently outside and her Noselength is below the dishonesty threshold
- A user enters the probationary period, if she is currently outside and her Noselength is above the dishonesty threshold
- A user stays in the probationary period, if she is currently inside and the probationary period has not passed yet, or if her Noselength is above the honesty threshold
- A user leaves the probationary period, if she is currently inside and her Noselength is below the honesty threshold and has been there for the whole probationary period

The Noselength value is also used by the system for classifying users into different categories. On one end of the spectrum, *radicals* are users who disagree with others much more often than other users. The other extreme class is *followers*, which consists of users who disagree the less often with others. Those users that maintain a healthy level of disagreement amongst them fall into the *average* class.

**Consultation to the RMS.** The reputation management system contacts Jiminy to query the trustworthiness of a certain user, in order to determine whether to reward her for submitting statements. A trustworthy user receives a credit amount for each statement she has submitted since the last query. The RMS interface during a query to Jiminy and the result returned by Jiminy are shown in Figure 4.

**Use of Classification Information.** The class in which a user resides — in terms of rating behaviour — can be obtained from Jiminy. This can be used to, for instance, transparently filter ratings taken into account for a given user. As



**Fig. 4.** Interface of the RMS for querying Jiminy

an example, a retailer web site linked to Jiminy could automatically calculate the average rating of a product based on the ratings of users in the same class as the user who is viewing the product. A radical user is likely to prefer seeing the ratings of other radical users, rather than those of sheep users.

## References

1. A. Fernandes, E. Kotsovinos, S. Ostring, and B. Dragovic. Pinocchio: Incentives for honest participation in distributed trust management. In *Proc. 2nd Intl Conf. on Trust Management (iTrust 2004)*, Mar. 2004.
2. E. Kotsovinos, P. Zerfos, N. Piratla, N. Cameron, and S. Agarwal. Jiminy: A Scalable Incentive-Based Architecture for Improving Rating Quality. In *Proc. 4th Intl. Conf. on Trust Mgmt (iTrust '06)*, May 2006.