Reputation systems

Keywords : Reputation; Rating; Filtering.

Cristobald de Kerchove, Paul Van Dooren

***Abstract** – We present a new filtering technique to correct the votes of raters that are possibly spamming or trying to cheat a particular reputation system, such as Amazon, Trip Advisor or Ebay.*

The World Wide Web is making more and more use of interactive ratings collected from various users: books are being evaluated in Amazon, movies are being rated in Movielens, web-users are even being judged in Ebay. This clearly is a form of voting but not all raters can be expected to be fully reliable or even honest. A rater on the Movielens database may give random ratings to movies he has not even seen, or a dishonest voter may give biased opinions just to favor his or her friends. From a commercial point of view, it is clear that web sites have a lot to earn by promoting confidence in such interactive rating systems. Ideally this would be achieved by penalizing raters that give random or biased ratings. Two questions ought to be addressed in this context: 1. What should be the reputation of the evaluated items? 2. How can we measure the reliability of the raters?

A natural way of tackling the problem of unreliable or unfair raters in reputation systems is to weight the evaluations of the raters. Hence the range of weights corresponds to a continuous scale of validation of the votes. These weights change via an iterative refinement that is guaranteed to converge to a reputation score for every evaluated item and a reliability score for every rater. At each step the reliability of a rater is calculated according to some distance between his given evaluations and the reputations of the items he evaluates. This distance is interpreted as the belief divergence. Typically, a rater diverging too much from the group should be distrusted to some extent. The same definition of distance appears in [1] and is used for the same issue. The strength of the new reputation system we describe in [2], is that it can be applied to any static network of raters and items and that it then convergences to a unique fixed point. Moreover, it can also be extended to dynamical systems with time-dependent votes.

We illustrate this method on an experiment involving a data set (supplied by the GroupLens Research Project) of 100,000 ratings given by 943 users on 1682 movies. The votes were ranging from 1 to 5. In order to test the robustness of our reputation system, we added 237 random raters to the original raters. In that manner, 20% of the users give random evaluations. Figure 1 gives the original (ordered) votes of the honest voters in blue and the (unfiltered) modified votes in green, when the spammers are also taken into account. Figure 2 gives the effect of the filtering operation on the same data, which clearly shows that the effect of the spammers has significantly been reduced. Figure 3 shows the distribution of the rater's weight as a result of our filtering operation: the random voters are seen to be generally penalized by a much smaller weight. Notice that we could have used this to remove the group of users considered as outliers, but our reputation system prefers to provide a continuous range of rates, instead.

The main issues of this new reputation system are: the relevance of the measure, the robustness against different sort of attackers, the application of the method for any sort of data and the ease of understand the measure by users.

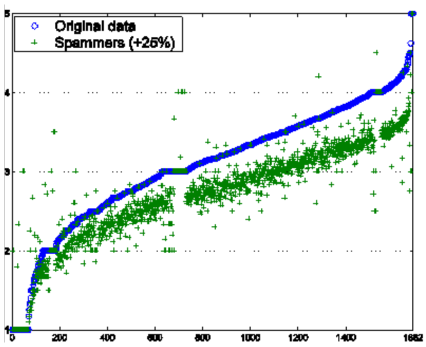

Figure 1: Results with average votes. Original votes in blue, spammed votes in green.

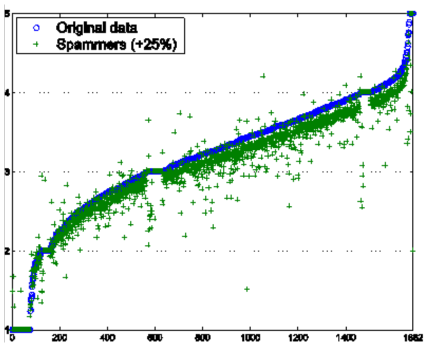

Figure 2: Results with filtered votes. Original votes in blue, spammed votes in green.

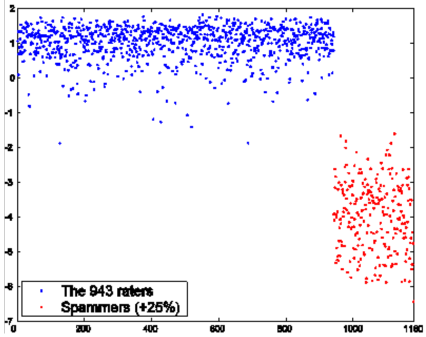

Figure 3: Filtered reputations of voters. Honest voters in blue, spammers in red.

References

[1] P. Laureti, L. Moret, Y.-C. Zhang and Y.-K. Yu, "Information Filtering via Iterative Refinement", Europhys. Lett. pp.1006-1022, 2006.

[2] C. de Kerchove and P. Van Dooren, "Iterative filtering for a Dynamical Reputation System", SIAM Matr. Anal. & Appl., pp.1812-1834, 2010.