The Influence Limiter: Provably Manipulation-Resistant Recommender Systems

Paul Resnick
University of Michigan
School of Information
presnick@umich.edu

Rahul Sami
University of Michigan
School of Information
rsami@umich.edu

ABSTRACT

An attacker can draw attention to items that don't deserve that attention by manipulating recommender systems. We describe an influence-limiting algorithm that can turn existing recommender systems into manipulation-resistant systems. Honest reporting is the optimal strategy for raters who wish to maximize their influence. If an attacker can create only a bounded number of shills, the attacker can mislead only a small amount. However, the system eventually makes full use of information from honest, informative raters. We describe both the influence limits and the information loss incurred due to those limits in terms of information-theoretic concepts of loss functions and entropies.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

General Terms

Algorithms, Reliability

Keywords

Recommender systems, manipulation-resistance, shilling

1. INTRODUCTION

Content posted on the Internet is not of uniform quality, nor is it equally interesting to different audiences. Recommender systems guide people to items they are likely to like, based on their own and other people's subjective reactions. We will refer to people's opinions generically as ratings, whether users explicitly enter them in the form of ratings or tags, or whether the system infers them from implicit behavioral indicators such as purchases, read times, bookmarks, or links.

Authors and other parties often want to direct attention to particular items. Google, Yahoo!, and others channel this into a multi-billion dollar advertising marketplace. But to the extent that people rely on recommender systems of various kinds to guide their attention, there are also natural incentives for promoters to manipulate the recommendations. An attacker may rate strategically rather than honestly and may introduce multiple entities, sometimes called *sybils*, to rate on behalf of the attacker.

We offer a manipulation-resistance algorithm, called the Influence Limiter, that can be overlaid on existing recommender algorithms. Consider the predictions about whether a particular target person will like various items. Each rater begins with a very low but non-zero reputation score. The current reputation limits the influence she[1] can have on the prediction for the next item. Eventually, the target person indicates whether he likes the item and the raters who contributed to predicting whether the target would like it gain or lose reputation. The more that a rating implies a *change* in the prediction for the target, the greater the potential change in the rater's reputation score. A rater who simply goes along with the previous change will have no impact and thus get no change in her reputation.

The Influence Limiter has several desirable properties. First, in order to maximize the expected reputation score of a single rater endowed with some information about the target's likely response to the items, the optimal strategy is to induce predictions that accurately reveal that rater's information about the items. If the underlying recommendation algorithm is making optimal use of ratings, this implies that entering honest ratings is optimal. An important special case is that a rater who has not interacted with an item, and therefore has no information about the target's likely response to it, can only lose reputation in expectation by giving a rating.

Second, the actual reputation score of any rater is always positive and is bounded above by an information-theoretic measure of the actual improvement that rater has made in the predictions for the target. It is not possible to prevent all manipulations of the predictions for particular items– a rater who provides good information on all other items but strategically provides bad information on one item is indistinguishable from a rater who simply has an unusual opinion on the item in question. Our algorithm does, however, ensure that no rater can negatively impact the overall set of recommendations for a target by more than a tiny amount. Moreover, if the item being manipulated is unlikely to be of interest to the target, later raters may provide information

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.
Copyright 2007 ACM 978-1-59593-730-8/07/0010 ...$5.00.

[1] We refer generically to raters as female and the target for predictions as male.

that corrects the prediction on that item before the target is affected by the recommendation.

Finally, our algorithm limits the amount of damage that can be done with sybils. For example, if one rater provides bad information about an item in order to increase the reputation of another rater who later corrects that bad information, the expected sum of the reputations of the two raters does not increase. Thus, while it may be possible to transfer reputation among sybils, it is not possible to increase the total reputation of the raters that a person controls. We presume that the recommender system imposes some minimal cost (or inconvenience) on the creation of rater entities. Thus, there is some bound (say, 1,000) on the number of sybils one person can create without it being too costly and without being detected. The initial reputation of each rater is set low enough that the total reputation of this bounded number of raters is still relatively small.

To further motivate our manipulation-resistance algorithm, consider some approaches to manipulating conventional recommender systems. One threat is a *cloning* attack. For example, in a recommender system that asks each rater to report movie ratings on a 1-5 scale, the attacker simply reports the same ratings as some other rater, except for a single item to be manipulated. Most recommender systems do not take into account the order of ratings, and thus the attacker will have just as much effect on predictions for the last item as the rater who was copied. In a nearest-neighbor recommender algorithm, the attacker can even just clone the ratings of the target; the attacker will then be the nearest neighbor of the target. The cloning attack can be made more difficult by hiding the actual rating vectors of raters, but significant information about others' ratings will leak out in the content of the recommendations, and sophisticated attackers will be able to create influential rating profiles through approximate cloning. Our approach thwarts the cloning attack by adding reputation only when a rater *improves* the prediction made for some target. Unless the rater moves the recommendation from where it was before the rater provided its information, the rater can neither gain nor lose reputation.

A second threat to conventional recommender systems comes from *random profile flooding*. An attacker creates a large number of sybils that provide random reports except on the item or items to be manipulated. By chance, some sybils may appear to have provided useful information in their random reports. These sybils are used to impact the prediction for an item being manipulated. Our algorithm thwarts this attack by making the probability of gaining sufficient credibility through random reports very low, so low that an attacker gains less influence in expectation from its sybils making random guesses than from simply transferring the initial credibility of all its sybils to a main identity. Moreover, any sybil profile that does happen to gain a high credibility score has, by chance, moved the predictions for the target in a useful way, thus compensating for the lost utility from the subsequent manipulation.

The paper begins with an exploration of related work in section 1.1. Section 2 presents a model of the recommending process. Section 3 presents our algorithm. Section 4 provides formal statements of its manipulation-resistance properties. Section 5 presents information-theoretic bounds on the information loss due to influence limits. Section 6 discusses limitations and possible extensions.

1.1 Related Work

The possibility of sybil attacks on recommendation systems has been noted by O'Mahony *et al.* [22] and Lam and Riedl [15], who use the term "shilling attack". Through simulation, Lam and Riedl study versions of the cloning and random profile attacks on different recommender algorithms, and note that the effectiveness of the attack varies depending on the algorithm used. However, they do not address the development of a provably attack-resistant algorithm.

Several authors have suggested using statistical metrics on ratings to distinguish "attack" identities from "regular" identities, and eliminate the former [7, 23, 18]. Mobasher *et al.* [20] survey this literature and classify attack strategies. This approach is likely to lead to an arms-race where shillers employ increasingly sophisticated patterns of attack. To avoid this, our approach does not rely on identifying particular attack identities or specific attack strategies. O'Donovan and Smyth [21] suggest using accuracy information from multiple targets to judge credibility; it would be interesting to see if our scheme can be extended in this way.

Dellarocas [8] provides an algorithm that bounds the damage that attackers can do when they collectively provide less than half the ratings in the system and the honest ratings are normally distributed. Our approach succeeds much more generally, even in situations where only a tiny fraction of the ratings are honest, at the expense of greater information loss during the startup phase when raters are not yet credible.

Herlocker *et al.* [13] study a modification of a nearest-neighbor recommender algorithm that does not count a rater as a near neighbor until it has rated sufficiently many items in common with a target rater. This is similar in spirit to our influence-limiting approach, but we provide a limiting process that is grounded in information theory and provably resistant to manipulation.

We use proper scoring rules to elicit honest ratings. These were pioneered in the context of forecasting objective events like weather patterns [4]; Miller *et al.* [19] noted that scoring rules could be adapted to the recommendation setting by treating the target's rating as an objective outcome. Hanson [11] developed the *market scoring rule* as a mechanism for information markets. In this mechanism, a trader is rewarded with the score difference between her prediction and the previous prediction. This relative scoring rewards the first provider of information, and forms an essential component of our approach. We develop additional machinery to handle strategies involving sybils and bankruptcy.

Bhattarcharjee and Goel [3] suggest sharing the revenue of a ranking system with the raters. Using techniques similar to market scoring rules to determine the revenue shares, they argue that an attack on the system would be costly. In contrast, we do not require any real money transactions, and prove bounds on the damage that sybils can do.

We use the error score change as a natural measure of performance of the system and damage to the system. Rashid *et al.* [25] propose several other algorithm-independent measures of rater influence; unlike error score change, their measures do not consider the dynamic order of ratings. Separately, Rashid *et al.* [24] use entropy to analyze a different problem: choosing which items to ask new users to rate.

The literature on bounded-regret online learning deals with combining predictions from multiple forecasters and proving worst-case bounds on the error relative to the best predictor that could be chosen in hindsight (see Cesa-Bianchi and Lugosi [5] and references therein). Many online learning algorithms can also be viewed as schemes for betting on a sequence of events [26]. The influence-limiting algorithm can be interpreted as an online learning algorithm: our damage bound is a form of relative error bound. The online learning literature typically does not take the order of forecasts on a single item into account; our algorithm is tailored to this critical feature of the recommender setting.

Awerbuch *et al.* [2] study manipulation in a different model of the recommendation process: a user samples items and recommendations until he likes an item, at which point he recommends that single item to others. They present a sampling algorithm and prove bounds on the number of samples required, even in the presence of adversaries. Awerbuch and Kleinberg [1] describe an online learning scheme that is provably good for a generalization of this problem that incorporates time-varying preferences and recommendations.

Sybilproofness has been studied in the context of reputation systems by Cheng and Friedman [6]. Their mechanisms begin with a trust graph: expressions of trust by raters about *other raters*. Advogato [16] and Eigentrust [14] also attempt to address manipulation in a trust-graph model. Massa and Bhattacharjee [17] show how external trust relationships can be used to improve recommender systems. These techniques are not directly usable in our setting, because raters may not know anything about the other raters in the system. In fact, our algorithm can be viewed as a way to securely derive a trust (or credibility) graph from ratings on inanimate objects.

2. THE RECOMMENDING PROCESS

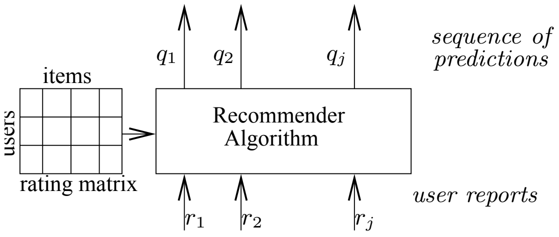

Figure 1: A recommender system

We now present a formal model of the recommending process, illustrated in Figure 1. Let N denote the total number of identities in the system, and $\mathcal{N}$ denote the set of identities. We use M to denote an upper bound on the number of items that are available for rating, so that each item can be assigned an index in $\{1, \cdots, M\}$.

There is a space $\mathcal{L}$ of possible labels, which describes the classification of items by the target for whom the system makes predictions. The prediction space $\mathcal{X}$ is a set of probability distributions over the label space; a prediction $\mathbf{q} \in \mathcal{X}$ is thus a distribution over labels. Prior to the target providing a label, a sequence of raters provide ratings $r_1, r_2, \cdots$ that change the recommender system's assessment of the target's probability distribution over labels.

We describe the process assuming that all raters enter ratings into the same underlying recommender algorithm, which combines them with previous ratings to generate predictions; this matches the standard mode of operation of current recommender systems. The formal analysis extends to more general settings: raters could directly report probabilities, incorporating past information in any way they like. In practice, the recommender algorithm would also use each incoming rating to update predictions about other items and for more than just one target person. It is sufficient for our purposes to describe the predictions for a single target.

Existing recommenders typically predict a single number, which corresponds to the mean of the distribution over labels on a numeric scale (e.g., 3.8 on a 1-5 scale). To function in our scheme, such recommenders could be extended to report entire probability distributions (e.g., 30% chance of 5; 20% chance of 4; 50% chance of 3). Alternatively, intermediate labels and point predictions can both be interpreted as a mixture of ratings over the extreme points (*e.g.*, 3.8 is a 70% chance of 5 and 30% chance of 1). However, in the special case where there are only two possible labels, the mean (e.g., .7 on a 0-1 scale) completely determines the entire distribution. In the rest of this paper, we shall assume that the label space is simply $\{HI, LO\}$, and that a prediction expresses the mean, the probability of a HI label.

2.1 Measuring Error: Loss Functions

In this section, we describe the error measures we use for a single prediction. Later, in section 5.1, we will see that these error measures lead naturally to information-theoretic measures of a rater's expected contribution.

We begin by assigning a value to good recommendations. We do this by postulating that the target has a *loss function* $L(l, q)$, where $q \in [0, 1]$ is the recommendation (predicted probability of rating HI) that the target was given, and $l \in \{HI, LO\}$ is the target's ultimate label. One common choice is the *log-loss* function:

$$L(HI, q) \quad = -\log q; \quad L(LO, q) \quad = -\log(1 - q)$$

Although typically logarithms to base 2 are used, so as to measure entropy in bits, we use natural logarithms (logarithms to base e) throughout this paper. Note that the loss is 0 when the prediction is completely accurate (i.e., when there is no error in the prediction).

One drawback of the log-loss function is that it is unbounded as q tends to 0 or 1. We need to avoid the possibility of infinite (positive or negative) scores. One alternative is the *Quadratic loss function/scoring rule:*

$$L(HI, q) = (1 - q)^2; \quad L(LO, q) = q^2$$

This corresponds to using the expected squared error (the variance) as a measure of uncertainty, which has a long history in statistics. It also corresponds to the use of mean squared error as a measure of prediction accuracy in recommender systems that make predictions on a 0-1 scale. The quadratic loss function is clearly bounded by $[0, 1]$.

3. THE INFLUENCE LIMITER ALGORITHM

Figure 2 depicts an influence-limited recommender system. There are two key additional features. First, the prediction q_j output by the recommender algorithm passes through an influence-limiting process to produce a modified prediction $\tilde{q}_j$. The influence-limiting process generates $\tilde{q}_j$ as a weighted average of the prediction $\tilde{q}_{j-1}$ prior to incorporating j's rating and the prediction q_j using j's rating. The weighting of the two terms depends on the reputation R_j that j has accumulated with respect to the target. When $R_j \geq 1$, all weight is on q_j, *i.e.*, j has full credibility.

The second feature is a scoring function that assigns reputation to j based on whether or not the target actually likes the item. The amount of reputation that can be gained is again limited by the current reputation, and is selected so that the reputation will always be positive. It is also tuned so that honest raters reach full credibility after $O(\log n)$ ratings, to limit the amount information that is discarded in the influence-limiting step before a rater reaches full credibility. If a rater's entire reputation was risked on each rating, the probability of getting to full credibility from a sequence of ratings the target agrees with would be too small and the expected time to full credibility would be too high. The rate limiter β_j threads the needle, allowing the rater to risk more as her reputation rises, but not risk too much, as we prove in section 5. [2]

Figure 3 presents the algorithm more formally. Each rater begins with a tiny reputation $e^{-\lambda}$, small enough so that even a large number of raters would have total reputation much less than 1. $\beta_j = \min(1, R_j)$ determines j's influence limit; if R_j is above 1, j can move the prediction $\tilde{q}_{j-1}$ to q_j, but if R_j is below 1 she can only move it partway there. The score for rater j on the current item is a fraction β_j times the market scoring rule, the difference between the loss function for the prediction $\tilde{q}_{j-1}$ and the loss function for q_j. Note that she is scored on the prediction she would have liked to make, q_j, even if the rate limiter only permitted her to move the prediction to $\tilde{q}_j$.

The log loss and quadratic loss functions are *proper scoring rules* [4, 9]. That means that its expected value is maximized when q_j matches the true probability that the target likes the item. This eliminates any incentive for rater j to lie about her rating, if she wants to maximize her expected reputation score, a property that we prove more formally in section 4. Note that honest reporting only maximizes the score in expectation; given the target's true opinion of the item, the actual loss is minimized with a prediction of that outcome with certainty.

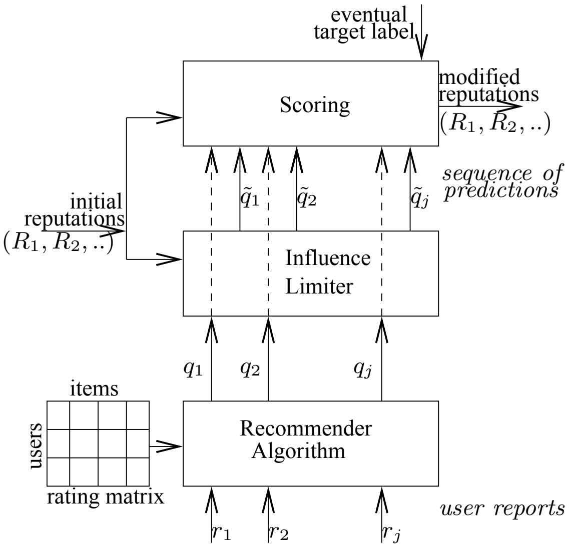

Figure 2: An influence-limited recommender system

```
ComputeReputations($\lambda$)
  1. Initialize $R_j = e^{-\lambda}$ for all $j$.
  2. For an item the target will eventually label do:
      a. $\tilde{q}_0 = p_0$
      b. Consider the ratings on the item in temporal order
      c. For each rater $j$:
      d.   $\beta_j = \min(1, R_j)$
      e.   $\tilde{q}_j = (1 - \beta_j)\tilde{q}_{j-1} + \beta_j q_j$
      f.   After the target provides label $l$, $R_j = R_j + \beta_j[L(l, \tilde{q}_{j-1}) - L(l, q_j)]$
```

Figure 3: Algorithm to compute reputations and limited predictions. Loss function L can be any proper scoring rule bounded by $[0, 1]$.

4. STRATEGIC PROPERTIES

Our main non-manipulation result shows that, for any attack strategy involving up to n sybils, the net negative impact (damage) due to the attacker is bounded by a small amount. We first state two important assumptions we make about the attack strategy, and then prove the results on damage limits. We also show that a user seeking to maximize her influence has a strict incentive to rate honestly.

Assumptions: One important assumption we make is that there is a number n that bounds the maximum number of fake identities a single attacker can create. As stated in section 1, we intend n to be a fairly large number, perhaps in the thousands or even millions. It might appear that limiting a user to a million identities is qualitatively as difficult as limiting her to one, but there is an important difference in practice. For example, many sites require new users to solve CAPTCHAs [27], which are puzzles that require a few seconds of human attention. While this is a mild inconvenience for most users who create only one account (or a few accounts), it can make creating millions of accounts prohibitively expensive. On the other hand, the only realistic way to limit a user to only one account is to use and verify personally identifying information for each account, which is a major privacy threat, and can be costly and time-consuming. Note that we do not assume that a constant fraction of the identities in the system are real users; the set of identities can be dominated by an attacker's sybils.

Secondly, we restrict our attention throughout this paper to *myopic* attack strategies. In particular, we do not consider reputation-building strategies in which the attacker enters poor ratings that mislead later raters into amplifying her misinformation, and then later corrects it with an additional rating from another sybil. This is a nontrivial assumption: In many cases, the optimal prediction after honest ratings from agents $1, 2, .., j$ will be sensitive to each of their inputs and might amplify the change in predictions by earlier raters.

[2] By analogy, consider a sequence of coin flips where you double your money on heads and give back all but \$1 on tails. The expected time to reach a bankroll of \$$x$ would be quite long, even if heads occurred with probability 0.75. By contrast, if tails only required giving back half your money, the expected time would be $2 \log x$.

These non-myopic strategies, if they exist, would be equivalent to manipulative strategies in information markets, analogous to initiating a buying frenzy for a particular stock and then selling to the very buyers who entered the market because of your actions. While there are market settings in which non-myopic strategies are theoretically possible, several studies have shown that such manipulation attempts are not very successful in practice (see, *e.g.* [12]). Non-myopic manipulative strategies, if they exist, are also likely to be quite complex for an attacker to carry out, as they would require delicate calculations about other agents' inference and learning methods. In this paper, we restrict our attention to preventing the simpler (but still very powerful) myopic attacks. Research on non-myopic manipulation in information markets, and methods to guard against them, will have direct relevance to our mechanism.

Impact of rater j: We now introduce a measure of rater j's impact on the recommendation quality. j's rating q_j on an item i has an immediate impact on the recommendations: The recommendation $\tilde{q}_{j-1}$ is changed to the recommendation $\tilde{q}_j$. We define the *myopic impact attributed to j for item i, Δ^i_j*, by:

$$\Delta^i_j \overset{def}{=} L(l^i, \tilde{q}_{j-1}) - L(l^i, \tilde{q}_j)$$

(Here, l^i is the target's rating on item i.) In other words, Δ^i_j measures the reduction in prediction loss that resulted from the immediate changes in recommendation following j's rating.

Note that myopic impacts are additive. The total impact of a sequence of raters is the reduction in error from the initial default prediction to the final prediction $\tilde{q}_J$. This can be expressed as the sum of the impacts attributed to the individual raters because the terms telescope:

$$\sum_j \Delta^i_j = \sum_j [L(l^i, \tilde{q}_{j-1}) - L(l^i, \tilde{q}_j)] = L(l^i, p_0) - L(l^i, \tilde{q}_J)$$

Finally, we define the *net impact* of rater j by the sum of her myopic impacts on all items i: $\Delta_j = \sum_i \Delta^i_j$.

We now prove our non-manipulation results in this context. First, we make a straightforward but important observation about the reputations calculated by the algorithm:

Observation: $R_j \geq 0$

This is true initially, and it is clearly maintained in step 2.f of the algorithm because $L()$ is bounded between 0 and 1.

LEMMA 1. *(Honest Reporting) For any entity j, if j believes (at the time of rating) that the probability of the item being labelled HI is q, j optimizes her expected reputation gain on this item by putting in a rating such that $q_j = q$.*

PROOF. From line 2.d, it can be seen that the change ΔR_j in j's reputation is proportional to $L(l_i, \tilde{q}_{j-1}) - L(l_i, q_j)$. Agent j cannot control the first term. As the scoring rule is proper, the second term is minimized in expectation by reporting $q_j = q$. □

Remark: In itself, this is an incentive for honest rating only if users care about their reputations or influence. Inducing users to care about reputations is an orthogonal issue; this might be done through status rewards for raters with high reputation, or by adding a little noise to the recommendations for users with low reputation.

Remark: If the recommender algorithm that generates the predictions q_j from the ratings $r_1, \cdots, r_j$ is imperfect, rater j may benefit by compensating for the algorithm: reporting a value r_j different from what she truly perceived in order to improve the prediction.

The following critical lemma relates the influence an agent garners to her measured impact on prediction error.

LEMMA 2. *(Reward-performance inequality) The reputation increase ΔR_j of player j on an item i is no more than the impact Δ^i_j of entity j on this item's recommendation.*

PROOF. Let β_j denote the influence limit calculated in step 2.d of the algorithm and l denote the eventual label on this item. Then, the reputation earned by player j for this item is

$$\Delta R_j = \beta_j [L(l, \tilde{q}_{j-1}) - L(l, q_j)]$$

The reduction in prediction error, on the other hand, is given by

$$\begin{aligned} \Delta^i_j &= L(l, \tilde{q}_{j-1}) - L(l, \tilde{q}_j) \\ &= L(l, \tilde{q}_{j-1}) - L(l, (1 - \beta_j)\tilde{q}_{j-1} + \beta_j q_j) \end{aligned}$$

(Note that ΔR_j and Δ^i_j may be negative.) For both scoring rules we consider, log and quadratic, the loss functions are convex, and so we have:

$$L(l, (1 - \beta_j)\tilde{q}_{j-1} + \beta_j q_j) \leq (1 - \beta_j)L(l, \tilde{q}_{j-1}) + \beta_j L(l, q_j)$$

Thus, substituting in the expression for Δ^i_j, we get

$$\Delta^i_j \geq \beta_j [L(l, \tilde{q}_{j-1}) - L(l, q_j)] = \Delta R_j \quad \square$$

COROLLARY 3. *At any point of time, the total myopic impact Δ_j of all ratings by entity j satisfies $\Delta_j \geq -e^{-\lambda}$.*

PROOF. From the algorithm, it is clear that β_j and R_j are never negative. Thus, the net decrease in reputation of entity j is at most $e^{-\lambda}$. By lemma 2, the net increase in prediction error $\Delta_j = \sum \Delta^i_j$ due to j's ratings is no more than this. □

We can now state our main manipulation resistance property:

THEOREM 4. *(Limited Damage) Consider any strategy for the attacker k; the strategy involves the creation of up to n sybils $K = \{k_1, k_2, \ldots, k_n\}$. Then, the total myopic impact of all of k's sybils is bounded by*

$$\sum_{k_t} \Delta_{k_t} \geq \sum_{k_t} R_{k_t} - ne^{-\lambda} \geq -ne^{-\lambda}$$

PROOF. It follows from Corollary 3 by simply summing over all sybil identities. □

Remark: Setting $\lambda = \log(cn)$ for some parameter c, Theorem 4 shows that an attacker cannot cause a total reduction in the system performance of more than $1/c$. This damage bound does not require any assumptions about rationality, prior probabilities, or honest ratings.

Note, however, that we are relying on the assumption that strategies that involve misleading other raters are impossible to carry out: otherwise, Δ_j would not completely capture the effect j has had on the recommendations. If one sybil can induce subsequent raters to amplify its effect on the predicted probability, then a later sybil may be able to secure an inflated Δ score, by correcting the amplified misinformation. The sum of all impacts still correctly measure the total reduction in prediction but the attacker would be stealing some impact score from the other raters who were misled by the initial sybil.

5. BOUNDING THE INFORMATION LOSS

Manipulation-resistance in itself is not very hard to achieve: simply ignoring all user ratings would result in nonmanipulable recommendations. It is therefore necessary to show that the recommendation scheme is still *informative.* In particular, the influence-limited operation does not make full use of the information from a user with low reputation. In this section, we show that users will in expectation require $O(\lambda) = O(\log n)$ ratings to build up their influence to an adequate level. We use this to bound the total information loss. We work primarily with the quadratic loss function, but the results should generalize to other loss functions with different constants.

5.1 Partial Information Model

In order to analyze the performance of the system in terms of reducing uncertainty, we need a formal model that captures the uncertainty of rating, raters' partial information signals, and raters' ability to learn from (and copy) previous ratings. We use a standard model of information partitions, which is described below. This model is used purely to analyze information loss; the operation of the Influence Limiter algorithm does not depend on it.

There is a space Ω of possible states (item types). For example, each state in Ω might correspond to a particular combination of a large number of movie attributes. For each $\omega \in \Omega$, there is a corresponding value $l(\omega) \in \{HI, LO\}$ that describes whether the target will like items in that state (*i.e.*, with that combination of attributes). Further, we assume there is a common prior probability distribution $p : \Omega \to [0, 1]$ that defines the relative likelihood of different states in the absence of additional information. We use p_0 to denote $P_p(l(\omega) = HI)$, the prior probability, before taking into account any ratings, that a randomly chosen item will be labelled HI by the target. We assume that the state space is finite.

When a rater acquires a signal about an item, we model her as eliminating some of the possible states, but not changing the relative likelihood of the remaining states. For example, she might be able to tell if the movie is violent or not, but not be able to discern some other characteristic, such as humor, that would distinguish among violent movies. More formally, all information acquisition is modelled in terms of identifying a component of a partition π of the state space Ω. A rater j has a partition $\pi_j = \{\pi_j^1, \cdots, \pi_j^t\}$. This is a partition, so $\pi_j^s \cap \pi_j^k = \emptyset$ for $s \neq k$, and $\cup_s \pi_j^s = \Omega$. The interpretation is that after acquiring a signal about the item, e.g., by watching the movie, j will know which component π_j^s of her partition the true state belongs in, but will not be able to distinguish two states in the same partition.

Given the loss function L, a natural way to frame the objective of the recommender system is to attempt to minimize the total error in the predictions, as measured by the sum of the losses. If no partial information from raters was available, the best prediction the system (or the target herself) could make would be to say $q = p_0 = P_p(l(\omega) = HI)$. This value of q can be shown to minimize the expected loss in the absence of any other information; the expected loss on each item is then given by the *entropy H(l)*: $H(l) = -p_0 L(HI, p_0) + (1 - p_0)L(LO, (1 - p_0))$.

The log-loss function (or *logarithmic scoring rule*) allows us to frame our results in terms of standard information-theoretic entropy. Other loss functions can be used as scoring rules in our algorithm, and would correspond to other measures of entropy, relative entropy, etc. (see the article by Grunwald and Dawid [10] for more information). The entropy measures thus derived are conceptually similar to the standard entropy.

Now, consider a sequence of raters $1, 2, \cdots, j$; each rating identifies a component of that rater's partition. Then, the prediction made after j's rating will depend on j's rating as well as all previous ratings. The information available to the recommender algorithm can be represented by a partition $\hat{\pi}_j$ that is a refinement of π_j, reflecting the fact that it can distinguish between items which j rated differently as well as possibly some items which were in the same component of π_j, but different components of the partition of some π_k for $k < j$. An ideal recommender algorithm then makes the best possible prediction q_j given this information.

A sequence of ratings by raters $1, 2, \ldots, j$ thus defines a sequence of partitions $\hat{\pi}_1, \cdots, \hat{\pi}_{j-1}, \hat{\pi}_j$. Generally, $\hat{\pi}_j$ is a *refinement* of $\hat{\pi}_{j-1}$: no component of $\hat{\pi}_j$ overlaps multiple components of $\hat{\pi}_{j-1}$. [3]

We can now define the innate *informativeness* of player j as the expected reduction in loss due to player j's participation as the jth rater in the sequence, before we learn what any of the realized ratings are. For any component $s \in \hat{\pi}_j$, define $s_{j-1} \in \hat{\pi}_{j-1}$ as the component of $\hat{\pi}_{j-1}$ containing s. We can then define $q_j(s) = P(l(\omega) = HI|\omega \in s)$ and $q_{j-1}(s) = P(l(\omega) = HI|\omega \in s_{j-1})$. That is, $q_j(s)$ is the posterior probability the target will like the item if the state is in the partition s, and $q_{j-1}(s)$ is the predicted probability after the rating sequence that would occur when the state is in s, but before taking into account j's rating.

The informativeness (*i.e.*, expected error reduction) $I(\hat{\pi}_j||\hat{\pi}_{j-1})$ due to the addition of user j's private information is calculated as:

$$I(\hat{\pi}_j|\hat{\pi}_{j-1}) \overset{def}{=} \sum_{\omega \in \Omega} p_\omega [L(l(\omega), q_{j-1}(\omega)) - L(l(\omega), q_j(\omega))]$$

$$= \sum_{s \in \hat{\pi}_j} p_s D(q_j(s)||q_{j-1}(s))$$

where $D(q||r) = q[L(HI, r) - L(HI, q)] + (1 - q)[L(LO, r) - L(LO, q)]$ is the *relative entropy*, a central construct in information theory. We can formally extend this definition to define $I(\mathbf{q}||\mathbf{u})$ for any functions $\mathbf{q}(\omega)$ and $\mathbf{u}(\omega)$ that are constant on components s, such as the influence-limited predictions $\tilde{q}_j$.

This $I(\hat{\pi}_j|\hat{\pi}_{j-1})$ is our measure of the incremental value of j's information given the ordering $1, 2, \cdots, j$. If j rated earlier in the sequence, she might look more informative, as her ratings might be less redundant with the information provided by previous raters.

5.2 Analyzing the information loss

In order to simplify the analysis, we assume a fixed rating order $1, 2, \ldots, j, \ldots$ on all items. We also assume that the ratings are reported without error and the underlying recommender algorithm correctly processes them to determine q values. These assumptions allow us to present a concise

[3] It is possible that $\hat{\pi}_{j-1}$ has two or more components in which the expected value of l is exactly the same, and hence, $j - 1$ rates in exactly the same way. However, as q_{j-1} takes the same value on all these components of $\hat{\pi}_{j-1}$, we can think of them as one component without altering the analysis.

information-theoretic bound on the information loss *due to influence limiting*. Note that there may be other sources of information loss in practice (*e.g.*, rating errors or recommender imperfections). As before, we assume that the ratings on different items are independent.

Given the order in which users rate, we want to relate the influence j earns to the informativeness $I(\hat{\pi}_j||\hat{\pi}_{j-1})$ of j's private information. From step 2.f of the algorithm, the expected growth in reputation of a rater j with reputation $R_j \geq 1$, whose rating identifies a component of $\hat{\pi}_j$ on which the prediction changes from u to q, is:

$$q[L(HI,u) - L(HI,q)] + (1-q)[L(LO,u) - L(LO,q)]$$

This is just the relative entropy, $D(q||u)$.

The players' reputations are initially very small. Initially, then, the amount that a reputation grows is scaled by R_j. We now show that a reputation builds exponentially in the number of ratings: The expected logarithm of the rater's reputation grows at a rate that depends on the rater's informativeness but not on the reputation.

Suppose a player with reputation $R_j < 1$ moves a prediction (on some item i) from u to q. With probability q, this item will eventually be labelled HI, and her reputation will be adjusted to $R'_j = R_j + R_j(L(HI,u) - L(HI,q))$. Similarly, with probability $1-q$, her reputation will be changed to $R'_j = R_j + R_j(L(LO,u) - L(LO,q))$. Then, the expected value of $\log R'_j$ is given by:

$$\begin{aligned} E(\log R'_j) &= q\log[R_j(1 + L(HI,u) - L(HI,q))] \\ &+ (1-q)\log[R_j(1 + L(LO,u) - L(LO,q))] \\ &= \log R_j + GF(q||u), \end{aligned}$$

where the *growth factor* $GF(q||u)$ is defined as

$$\begin{aligned} GF(q||u) &\overset{def}{=} q\log(1 + L(HI,u) - L(HI,q)) \\ &\quad +(1-q)\log(1 + L(LO,u) - L(LO,q)) \end{aligned}$$

The growth factor determines the expected rate of growth of j's reputation a single component of $\hat{\pi}_j$ that j' rating identifies. We define an *expected growth factor* measure $EGF(\hat{\pi}_j||\hat{\pi}_{j-1})$ by averaging over all possible components that j's rating could identify:

$$EGF(\hat{\pi}_j||\hat{\pi}_{j-1}) \overset{def}{=} \sum_{s\in\hat{\pi}_j} p_s GF(q_j(s)||q_{j-1}(s))$$

As in the case of the informativeness measure, we can extend this definition naturally to define $EGF(\mathbf{q}||\mathbf{u})$ for any functions $\mathbf{q}(\omega)$ and $\mathbf{u}(\omega)$ that are constant on components of $\hat{\pi}_j$ and $\hat{\pi}_{j-1}$ respectively.

Lemma 5 shows that the growth factor on any single component of $\hat{\pi}_j$ is close to the relative entropy on that component. Lemma 6 extends this to show that the expected growth factor is close to rater j's informativeness $I(\hat{\pi}_j||\hat{\pi}_{j-1})$, even if j is scored relative to the previous influence-limited prediction $\tilde{q}_{j-1}$ instead of the previous accurate prediction q_{j-1}. Proofs of these results and theorem 7 are omitted from the paper due to space restrictions; they can be found in an Appendix which has been archived online at *http://hdl.handle.net/2027.42/55415*.

Lemma 5. *For the quadratic scoring rule (MSE) loss, for all $q, u \in [0,1]$, $GF(q||u) \geq \frac{D(q||u)}{2}$.*

Lemma 6. *Suppose $\hat{\pi}_j$ and $\hat{\pi}_{j-1}$ are two partitions such that $\hat{\pi}_j$ is a refinement of $\hat{\pi}_{j-1}$. For each state ω, let $\mathbf{q}_j(\omega) = E(l(\omega)|\hat{\pi}_j)$ be the optimal prediction function given partition $\hat{\pi}_j$. Let $\mathbf{u}(\omega)$ be any function that is constant on each component of $\hat{\pi}_{j-1}$. Then, $EGF(\mathbf{q}_j||\mathbf{u}) \geq I(\hat{\pi}_j||\hat{\pi}_{j-1})/2$ in the quadratic loss model.*

Theorem 7. *Suppose rater j has rated m items, and suppose the informativeness of rater j is $I(q_j||q_{j-1}) = h$. Then, for all $m \geq (2\lambda + 1)/h$, rater i's expected reputation (with the quadratic scoring rule) is bounded below by*

$$E(R_j) \geq mh - 2\lambda - 2\log(mh - 2\lambda)$$

The intuition behind this bound is simple: While j's reputation is below 1, $\log R_j$ increases by at least $h/2$ in expectation; thus, after $2\lambda/h$ ratings, $\log R_j$ will be at least 0 (*i.e.*, R_j will be at least 1) in expectation. After this, j is not influence limited, and she will earn (an expected amount of) h in reputation in each subsequent round. Thus, in each of the last $(m - 2\lambda/h)$ rounds, she gains reputation h in expectation, for a total of about $(mh - 2\lambda)$.

An important consequence of Theorem 7 is that, by the Reward-Performance inequality (Lemma 2), the expected impact of player j will be greater than her reputation. In other words, we can expect to discard *at most* about 2λ units of information from each rater in total.

Note that this information loss bound employs a very conservative accounting scheme. Without the Influence Limiter, rater j would have contributed h bits on item i, but the influence limits reduces the effect of j's rating, we account for this as lost bits. However, if there is a subsequent rater $j+1$ with sufficient reputation, the eventual prediction may be the same as without influence limits. Rater $j+1$ might have been redundant in the absence of influence limits, but contributes valuable information to compensate for the loss of j's information when influence limits are present. Thus, the information loss bounds are based on a worst-case scenario.

6. CONCLUSION AND FUTURE WORK

We have presented an architecture and algorithm to limit the influence of individual raters that is provably manipulation-resistant (even when an attacker can create a large but bounded number of sybils), yet discards only a bounded amount of information from each rater. The smaller the size of the sybil attack that must be prevented, the less information that will be discarded from honest raters as they build up their credibility. There are two key ingredients to our system. First, we use the same metric L to measure the prediction performance of the system and to calculate raters' contributions (reputations). Second, the bootstrapping scheme is carefully balanced: we tie the influence a rater can have to her accumulated reputation in a way that enables informative raters to ramp up exponentially fast, while limiting the total damage at any point to the amount of positive impact already created by each rater.

Our results suggest several questions for future work. We have presented the algorithm as if one item is under consideration at any point of time, or equivalently, as if we get feedback about the true label as soon as the recommendation is used. In practice, several recommendations may be used before we get feedback about any of them; our algorithm will need to be modified slightly to manage "credit" in this scenario.

Another complication that we have not yet addressed is the target's selection bias in labeling items. The target is more likely to view items with high recommendations, and as a result, items with low recommendation values might be less likely to be labelled; this could influence the raters' incentives to provide effort. Modified scoring schemes that correct for this bias would be very interesting.

Our manipulation resistance result assumes that an attacker cannot mislead future raters into amplifying the effects of their ratings in a way that boosts the attackers' own credibility. Another area of interest for future work is to analyze and prevent such non-myopic strategies.

Although we have proven theoretical bounds on information loss, any amount of discarded information during the initiation period is still of concern, particularly in settings in which each individual rater has low incremental informativeness h or will rate only a few items. The algorithm will need to be carefully tuned to work well in such settings. A lower bound on the minimum information loss that any manipulation-resistant algorithm must incur would also be useful for comparison.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS 0308006.

7. REFERENCES

[1] B. Awerbuch and R. D. Kleinberg. Competitive collaborative learning. In *18th Annual Conference on Learning Theory (COLT 2005)*, volume 3559 of *LNAI*, pages 233–248. Springer, 2005.

[2] B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. R. Tuttle. Improved recommendation systems. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1174–1183. SIAM, 2005.

[3] R. Bhattacharjee and A. Goel. Algorithms and incentives for robust ranking. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, 2007.

[4] G. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[5] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[6] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132, 2005.

[7] P.-A. Chirita, W. Nejdl, and C. Zamfir. Preventing shilling attacks in online recommender systems. In *WIDM 05*, pages 67–74, 2005.

[8] C. Dellarocas. Building trust online: The design of robust reputation reporting mechanisms in online trading communties. In *Information Society or Information Economy? A combined perspective on the digital era*, pages 95–113. Idea Book Publishing, 2003.

[9] I. J. Good. Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114, 1952.

[10] P. Grunwald and A. Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *Annals of Statistics*, 32:1367–1433, 2004.

[11] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

[12] R. Hanson, R. Oprea, and D. Porter. Information aggregation and manipulation in an experimental market. *Journal of Economic Behavior and Organization*, page (to appear), 2006.

[13] J. Herlocker, J. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5:287–310, 2002.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Prooceedings of WWW '03*, pages 640–651, 2003.

[15] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of WWW '04.*, pages 393–402, 2004.

[16] R. Levien. *Attack-Resistant Trust Metrics*. PhD thesis, University of California, Berkeley, 2004.

[17] P. Massa and B. Bhattacharjee. Using trust in recommender systems: An experimental analysis. In *Proceedings of the 2nd International Conference on Trust Management*, 2004.

[18] B. Mehta, T. Hoffman, and P. Fankhauser. Lies and propaganda:detecting spam users in collaborative filtering. In *Proceedings of IUI'07*, 2007.

[19] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting honest feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[20] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*, 7(2):1–40, 2007.

[21] J. O'Donovan and B. Smyth. Trust no one: Evaluating trust-based filtering for recommenders. In *Proceedings of IJCAI'05*, 2005.

[22] M. O'Mahony, N. Hurley, and G. Silvestre. Promoting recommendations: An attack on collaborative filtering. In *Proceedings of the 13th International Conference on Database and Expert System Applications*, pages 494–503. Springer-Verlag, 2002.

[23] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre. Detecting noise in recommender system databases. In *Proceedings of the 2006 International Conference on Intelligent User Interfaces*, pages 109–115, 2006.

[24] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *IUI '02*, pages 127–134, 2002.

[25] A. M. Rashid, G. Karypis, and J. Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. In *Proceedings of the SIAM International Conference on Data Mining*, 2005.

[26] G. Shafer and V. Vovk. *Probability and Finance: It's Only a Game!* John Wiley and Sons, 2001.

[27] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *Proceedings of Eurocrypt 2003*, 2003.