JMLR: Workshop and Conference Proceedings vol 49:1–28, 2016

Provably Manipulation-Resistant Reputation Systems

Paul Christiano PAULFCHRISTIANO@GMAIL.COM
UC Berkeley

Abstract

Reputation and reliability play a central role in a wide range of applications, from online marketplaces to review aggregators to ridesharing services. Many reputation systems are vulnerable to manipulation, and protected only by keeping algorithms secret, avoiding high-stakes applications, or using side information to identify malicious users. The current situation is reminiscent of pre-modern cryptography, characterized by a patchwork of ad hoc techniques with minimal formal understanding of their security.

We propose a reputation system which provably achieves a very strong correctness guarantee under extremely pessimistic assumptions—it works even given a supermajority of malicious users, converges to optimal behavior after a constant number of interactions per user, does not require repeated interactions, and accommodates time-varying quality of resources.

Our formal model is simple but general. In each period, a user is given an opportunity to interact with a resource, and must accept or reject the proposed interaction. If they accept, they receive a payoff in $[-1, 1]$. Ideally all users would behave honestly, pooling their data and quickly learning which resources are worth interacting with. Our protocol essentially matches this performance when all users are honest, while guaranteeing that adding malicious users or users with varying tastes does very little damage.

We also extend our results to a more challenging setting where users interact with each other rather than with static resources, and where the two parties to an interaction may receive different payoffs.

Keywords: Online learning, collaborative learning, manipulation-resistance, collaborative filtering, reputation systems

1. Introduction

The Internet suffers from a fundamental tension: it provides access to an inexhaustible wealth of content, knowledge, and opportunity, but also exposes us to an inexhaustible supply of malice, incompetence, and greed. It seems all but impossible to realize one without the other. Compared to our local communities, the Internet is an unfriendly place full of unaccountable strangers, and we behave accordingly.

Simple solutions to this problem invite an arms race between honest users and malicious manipulators. If we use reviews to identify good merchants, bad merchants will pay for fraudulent reviews. If we use social networks to identify honest users, malicious users will find fake friends. These are not hypothetical concerns: there is ample evidence that manipulation is common across a very wide range of contexts. If we want to use reputation systems in high-stakes situations, they need to be robust to manipulation.

Identifying fraudulent reviews or malicious users by behavioral characteristics, such as their choice of words or the timing of their activities, is akin to "security by obscurity": when it works, it relies on the attackers' ignorance, incompetence, or fear of legal consequences. But is it possible

to do better? If manipulators can imitate honest users perfectly, is there any way to mitigate their influence?

We introduce a very simple and general model of interactions in an online community. In each period, two users have a chance to interact with each other, or a user has a chance to interact with a resource such as a movie or merchant.[1] If they choose to interact, then each user receives a payoff in $[-1, 1]$.

We propose a protocol for answering queries of the form "should user x interact with resource y?" or "should user x_0 interact with user x_1?"

We show that *every* set of users who follow our protocol achieves a total payoff nearly as high as if they had used the optimal whitelist/blacklist to decide which resources or users to interact with. This holds even when almost all users behave adversarially, when the quality of resources varies arbitrarily over time, when assessments are very noisy, and when the number of interactions per user is a constant independent of the size of the community.[2] This problem is difficult because we need to satisfy a regret bound for every set of users simultaneously—we are ignorant not only about what strategy will perform best, but also about whose payoff we are trying to maximize.

We must address two key technical challenges. The first challenge is the computational complexity of optimizing over an exponential space of possible filtering policies. This is addressed by applying recent algorithms for online matrix prediction based on semidefinite programming (Hazan et al. (2012); Christiano (2014)). The second challenge is to achieve low regret for every set of users simultaneously. Overcoming this challenge is key to ensuring robustness and non-manipulability. We propose a new algorithm which uses matrix prediction to iteratively refine a filtering policy, and analyze our algorithm using a novel application of traditional online learning guarantees.

When users interact with each other, we must address a third challenge: an interaction may be good for one user but bad for another. We need honest users to interact with each other whenever the *total* payoff is positive, But we also need to prevent honest users from interacting too much with "free riders" who receive high payoffs themselves but whose interaction partners receive low payoffs.[3] To address this challenge we introduce an accounting system based on logarithmic barrier functions. We show that this system effectively limits the extent of free-riding while allowing honest users to freely interact with each other.

These results suggest that reputation systems are not inherently vulnerable to manipulation, just as communication is not inherently vulnerable to eavesdropping. We raise a raft of open problems, from achieving statistically optimal and distributed protocols, to incorporating side information, to designing more complex and realistic systems that use trust as a building block.

In Section 2 we state our model. In Section 3 we sketch our approach to collaborative filtering. In Section 4 we present our complete algorithm for collaborative filtering and for reputation systems in the case of symmetric payoffs. In Section 5 we generalize this algorithm to the reputation system setting under a weaker "ex ante" symmetry condition.

1. The only difference between a "user" and a "resource" is that users receive payoffs.
2. It is clear that a set of users can only converge to this benchmark once $\Omega(1)$ of them have interacted with each resource: there is no way that the honest users can tell whether a given resource is good until at least one of them has interacted with it. We essentially match this lower bound, although we require ε^{-3} honest interactions in order to effectively estimate a resource's quality to within ε, while the statistical lower bound is ε^{-2}.
3. In our model, this would include e.g. a user who asks for favors much more often than they return them, or a user who often downloads files from a P2P service but rarely uploads them.

1.1. Related work

Many researchers have explored systems for collaboratively deciding what resources or users are worth interacting with. Our work is the first that is both *statistically efficient* and *robust to manipulation*: in the case of honest users our protocol reduces to an efficient online learning procedure, but it also guarantees that adversarial participants cannot significantly degrade the results.

Collaborative filtering: In the collaborative filtering problem, a set of users interact with a set of resources, and exploit their common tastes to more efficiently predict which resources each of them will rate highly. This problem has been studied at length; see (Su and Khoshgoftaar (2009)) for an overview.

In contrast with this literature, we focus on robustness and non-manipulability. Existing work makes essentially no guarantees if we include even a small fraction of users who behave manipulatively. Most existing work also makes strong assumptions on the relationship between the preferences of different users (such as the existence of an approximately low-rank decomposition), while we only assume that there exist sets of users who could benefit by pooling their information.

Some collaborative filtering systems, for example (Alon et al. (2006); Awerbuch et al. (2008)), make guarantees for every sub-community and are robust to adversarial manipulation. However, these systems aim to recover entire preference functions under relatively optimistic conditions, such as static preferences, strong similarity between the preferences of different users, and deterministic on-demand access to those preferences. We weaken these assumptions considerably, allowing weak similarities between users, time varying and noisy preferences, and adversarial patterns of available interactions.

Our results build on recent results in matrix prediction due to Hazan, Kale, and Shalev-Schwartz (Hazan et al. (2012)). Their algorithm can be applied directly to the collaborative filtering setting and is robust to many kinds of variation in preferences; but it cannot handle even mild manipulative behavior.

Manipulation-resistance: Another literature attempts to modify reputation systems to limit the influence of sybils, fake identities controlled by an attacker. For example, see (Yu et al. (2009); Resnick and Sami (2007)). This work focuses on a setting where a user can get useful feedback from the same peer across many different decisions, and so can directly learn about the reliability of each peer. In our setting only a handful of users have information about each decision, and so a user can rarely or never get direct advice from the same source more than once. Existing techniques do not achieve useful bounds in this setting.

Competitive collaborative learning: From a modeling perspective, our work is most similar to the competitive collaborative learning framework of Awerbuch and Kleinberg (Awerbuch and Kleinberg (2005)). In their model, honest users collaborate to discover an approximately optimal resource as quickly as possible. In contrast, in our setting finding the optimal policy requires making judgments about arbitrary resources. This significantly complicates the learning process: if the optimal policy depends on only one resource then it can spread between the honest users like gossip. In contrast, in our setting the honest users must learn to assess each other's reliability across a wide range of contexts.

Economic systems for P2P networks: Many decentralized economic protocols have been proposed to reduce the scope for free-riding and enforce reciprocity, especially in the context of Peer-to-Peer (P2P) networks; for some representative examples see (Cohen (2003); Meulpolder et al.

(2009); Vishnumurthy et al. (2003); Gupta et al. (2003); Seuken et al. (2010); Seuken and Parkes (2014)).

We are interested in addressing the problem of free-riding, which is also the motivation of most work in this area. However, we are interested in proving that our system works in our adversarial context. This requires us to focus on issues like price levels—how much money should a user be willing to pay for a given real benefit? —and liquidity—how do we ensure that users have enough money to facilitate honest transactions? These issues have not received much theoretical attention in the context of P2P networks, but are critical to proving that our system actually achieves its intended purpose.

2. Our model

2.1. Collaborative filtering: users interacting with resources

We consider a set $\mathcal{X}$ of users and a set $\mathcal{Y}$ of resources, such as blog posts, hotels, or merchants.

We will describe an algorithm that could be used by a centralized online service, making recommendations to individual users. It is also possible to implement our algorithm in a distributed setting, but we consider a centralized implementation for expositional clarity.

In each round $t = 1, 2, \ldots, T, \ldots$:

1. An adversary chooses $x^t \in \mathcal{X}$ and $y^t \in \mathcal{Y}$, and reveals them to the algorithm. For example, this may correspond to user x^t searching for an item in an online marketplace, identifying a merchant y^t selling that item, and then consulting the collaborative filtering system to determine whether to purchase from that merchant.

2. An adversary chooses a reward $p^t \in [-1, 1]$, but does not reveal it to the algorithm.

3. The algorithm outputs $s^t \in \{0, 1\}$, potentially stochastically. Outputting $s^t = 1$ indicates a recommendation to interact, while $s^t = 0$ indicates a recommendation not to interact.

4. If $s^t = 1$, then the algorithm observes p^t.

For any set of users $H \subset \mathcal{X}$, define $p^{\leq T}(H)$ to be the total payoff

$$p^{\leq T}(H) = \sum_{t \leq T : x^t \in H} s^t p^t.$$

Define $\text{OPT}^{\leq T}(H)$ to be the total payoff that users in H would have obtained over the first T rounds by choosing the optimal set of resources S and interacting only with resources from that set. That is, define:

$$\text{OPT}^{\leq T}(H) = \max_{S \subset \mathcal{Y}} \sum_{t \leq T : x^t \in H, y^t \in S} p^t.$$

The *regret* of the users in H is the difference between the benchmark $\text{OPT}^{\leq T}(H)$ and their actual payoff $p^{\leq T}(H)$.

In Appendix C we exhibit a polynomial-time algorithm $\mathcal{A}_{\text{filtering}}$ and prove:

Theorem 1 *For any set of users $H \subset \mathcal{X}$ and any $T > 0$, $\mathcal{A}_{\textit{filtering}}$ satisfies*

$$p^{\leq T}(H) \geq \textit{OPT}^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right)$$

where $N = |\mathcal{X} \cup \mathcal{Y}|$.

The sequences x^t and y^t, and the payoffs p^t, are all adversarial and may depend on the past behavior of the algorithm. This adversarial dependence captures arbitrary malicious behavior by the users and resources. We effectively model all users outside of H and all resources outside of S as controlled by a single adversary who knows everything except for the future random choices of the algorithm.

To get some intuition for the result, consider the case where $|\mathcal{Y}| = \Theta(|\mathcal{X}|)$. Let $k = T/|\mathcal{X}|$ be the average number of interactions per user, and let $\alpha = |H|/|\mathcal{X}|$ be the fraction of users in H. Then our regret is $\mathcal{O}\left(k^{2/3}\alpha^{-1}\right)$ per user in H. The regret would be $\mathcal{O}\left(k^{1/2}\alpha^{-1}\right)$ if the users in H knew each others' identities in advance and optimally pooled their information.[4]

Note that this guarantee holds for every set of users H simultaneously, and different subsets will have different optimal sets S. We can apply this analysis to whatever subsets give the most useful lower bounds.

Our algorithm solves a *filtering* problem—our protocol will simply decide whether proposed interactions should occur, rather than recommending the best interaction. Such a filtering system could be paired with a recommendation system if desired, such as proposing the cheapest merchant that hasn't yet been rejected, or proposing the interaction that is most likely to be accepted. Because our analysis holds in the worst case, we can apply it even when this filtering system is one component of a more complex recommendation system. The analysis of the composite system would have some additional complications: we raise this as an important open problem in Appendix F.

Finally, note that although the algorithm directly observes the payoff p^t for every interaction, the bound and benchmark only depend on the values of these payoffs for rounds when $x^t \in H$. Moreover, the bound holds even when the payoffs p^t are chosen arbitrarily. So we can safely run the algorithm using self-reported payoffs even in the presence of malicious users. If user x^t does not report a payoff in $[-1, 1]$, we take $p^t = 0$.

2.2. Reputation systems: users interacting with users

In the previous setting, each interaction involves a user x^t and a resource y^t. We can make the problem more challenging by allowing users to interact with other users.

In this setting, both parties to an interaction are users, $x_0^t, x_1^t \in \mathcal{X}$. As before, the algorithm outputs $s^t = 1$ or $s^t = 0$ according to whether it recommends interaction. If $s^t = 1$, the algorithm observes two payoffs $p_0^t, p_1^t \in [-1, 1]$.

The payoff of a set of users H is now defined by

$$p^{\leq T}(H) = \sum_{t \leq T : x_0^t \in H} s^t p_0^t + \sum_{t \leq T : x_1^t \in H} s^t p_1^t$$

We will modify the definition of our benchmark to impose an "ex ante" symmetry condition. Rather than having the adversary choose a pair of payoffs p_0^t, p_1^t directly, we assume that the adversary

4. In the lower bound k is actually the average number of interactions per user in H. We expect this to be comparable to the average number of interactions per user.

chooses a distribution over pairs of payoffs. We define

$$\text{OPT}^{\leq T}_{\text{sym}}(H) = \sum_{t \leq T : x_0^t, x_1^t \in H} 2 \min \left\{ \mathbb{E}\left[p_0^t\right], \mathbb{E}\left[p_1^t\right] \right\},$$

where the expectations are taken over the adversary's distribution.

To see why the "ex ante" symmetry condition is necessary, consider a simple example where users request and offer favors to one another, and use a reputation system to decide whether it is worth doing a favor for a peer. Our modified benchmark amounts to eliminating unreciprocated altruism: we assume that Alice has roughly as many opportunities[5] to help Bob as Bob has to help Alice.[6] Some assumption of this form, which excludes "free riders" from the set H, is clearly necessary: maximizing the welfare of free riders is equivalent to being exploitable by free riders, so we can't be both generous and non-exploitable.[7] This condition could be weakened by imposing a global balance condition rather than a pairwise balance condition, or by allowing unequal weights for different users; we leave these as important open questions.

We exhibit a polynomial time algorithm $\mathcal{A}_{\text{reputation}}$ and prove:

Theorem 2 *For any set of users $H \subset \mathcal{X}$ and any $T > 0$, $\mathcal{A}_{reputation}$ satisfies*

$$p^{\leq T}(H) \geq \mathit{OPT}^{\leq T}_{\mathit{sym}}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right),$$

where $N = |\mathcal{X}|$.

As before, note that this guarantee holds for any set of users. For example, we could take H to be the set of all users who never break a community norm, or the set of users who behave kindly or competently (for whatever definitions are relevant to a given application). Or we might want to apply the analysis to a smaller set H characterized by shared expectations or tastes.

Finally, note that our theorem does not rule out the existence of incentives for users to distort their reporting or to ignore our recommendations in specific cases. Our theorem applies regardless of how the users behave; so such distortions won't break our guarantee, but they might make it less meaningful. For example, if we let p_i^t be the payoff reported by x_1^t, and that report is dishonest, then maximizing $\sum s^t p^t$ may not be useful. We could apply our theorem with H containing only those users who report honestly, but if everyone reports dishonestly then that won't help either.

3. Sketch of the algorithm

In this section we will present the intuition behind our algorithm in the case of collaborative filtering. In section 4 we will give a formal presentation of the algorithm.

5. Of course in reality they may interact only a single time, this is only a statement about expectations.
6. In many settings, interactions are inherently inequitable and cash payments are used to ensure that both participants in an interaction benefit—e.g. the traveler will pay the host who provides them lodging. Our theorem applies regardless of the size of such a transfer, but the conclusion is strongest when the transfer is large enough that the two parties benefit equally (in expectation) from the transaction.
7. Formally, suppose that there are only two users, Alice and Bob. Suppose that every time they interact, Alice receives a payoff of -0.1, and Bob receives a payoff of $+1$. If Alice and Bob are both in H, then the naive benchmark would be $0.9T$, which could be attained only if Alice and Bob almost always interact. But if only Alice is in H, then any sensible benchmark is 0, which can be attained only if Alice and Bob almost never interact. In this case we would refer to Bob as a "free rider." No algorithm can possibly succeed in both cases, and it's not clear which is the "right" behavior.

3.1. Naive approach

Our first approach is to apply online learning to maximize the total welfare of all users:

- Consider all possible pairs $H \subset \mathcal{X}, S \subset \mathcal{Y}$. Each of these corresponds to a strategy for the filtering system: set $s^t = 1$ iff $x^t \in H$ and $y^t \in S$. Initially assign each of these strategies a weight of 1.
- In round t, pick a random strategy, with the probability of each strategy proportional to its weight. Follow the recommendation of this strategy.
- After observing p^t, adjust the weights of the strategies using multiplicative updates.[8]
- The resulting composite strategy receives a total payoff within $\mathcal{O}\left(\sqrt{TN}\right)$ of the best possible choice of H and S.

This will end up being the core of our approach, but it has two fundamental problems:

1. This algorithm guarantees that the total payoff of *all* users is close to the optimum, but it doesn't make any guarantee for the users in any particular set H.
2. There are exponentially many subsets, and so this algorithm is intractable.

To illustrate the first problem, suppose a malicious merchant creates a number of sybils (fraudulent accounts) and fabricates positive interactions with them. As long as the interactions with sybils are sufficiently positive, including the sybils in H and the dishonest merchant in S will increase the total payoff of the strategy corresponding to (H, S). But it will generally decrease the payoff of the honest users, by leading them to interact with the dishonest merchant. In a more extreme attack, dishonest users could create a large community of interacting sybils which is indistinguishable from the "honest" users of the system. A naive algorithm would simply maximize the welfare of the largest or most vocal community, potentially at the expense of all other users.

This is a fundamental challenge for reputation systems. Many existing reputation systems, such as EigenTrust (Kamvar et al. (2003)), are vulnerable to similar attacks. Straightforward adjustments, and in particular considering a richer space of strategies,[9] are not sufficient to address the problem.

In Section 3.3 we show how to address problem 1, by applying online learning to iteratively improve a filtering policy. This allows us to construct very complex filtering policies which cannot be significantly improved by any simple perturbation. We show that the resulting policies simultaneously minimize the regret of every set of users.

In Section 3.2 we show how to address problem 2 by using recent results in matrix prediction.

In Section 4 we present our full algorithm, integrating both of those ideas, and sketch the analysis.

8. Technically this is a contextual bandits problem, since we only observe the payoff p^t if we pick $s^t = 1$. But because there are only two options—0 or 1—this complication doesn't change the regret bounds.
9. We considered sets of strategies based on clusterings, divisions into overlapping cliques, and low-rank interaction matrices. None of these approaches can address the basic difficulty.

3.2. Matrix prediction

In order to avoid considering all 2^N pairs of subsets (H, S), we apply recent results from matrix prediction (Hazan et al. (2012); Christiano (2014)).

The first idea is to work with the space of matrices $P \in \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ with entries in $[0, 1]$. We view the entry P_{xy} as representing the probability that $x \in H$ and $y \in S$, for some hypothetical distribution over pairs (H, S). Each of these matrices corresponds directly to a strategy: recommend that x and y interact with probability P_{xy}.

If we could use online learning to make recommendations as good as the best matrix P, that would be even better than making recommendations as good as the best subsets (H, S). Unfortunately, this is impossible: without further assumptions on the matrix, each entry is essentially an independent prediction problem, and so we cannot converge to good behavior until every user has interacted with every resource.

We would be happy to only compete with the subset of matrices that actually arise from a distribution over pairs (H, S). Unfortunately, this set of matrices is very complex, and optimizing over it is intractable—it is a generalization of the Max-Cut problem.

The analogy with Max-Cut suggests a possible solution: we can instead optimize over the set of positive semidefinite matrices. That is, every matrix corresponding to an actual distribution over pairs (H, S) appears as the $\mathcal{X} \times \mathcal{Y}$ submatrix of a PSD matrix in $\mathbb{R}^{\mathcal{X}\cup\mathcal{Y}\times\mathcal{X}\cup\mathcal{Y}}$. (Christiano (2014)) presents an algorithm that achieves regret $\mathcal{O}\left(\sqrt{TN}\right)$ compared to the best matrix of this form, which is even better than competing with the best single pair (H, S).

This approach can address our computational difficulties. But it still leaves us with our second problem: the *total* payoff of all users will be as good as if we had chosen the best set (H, S), but the payoff of the honest users might be arbitrarily poor.

3.3. Iteratively adjusting the filtering policy

Our second key idea is using online learning to *update* the matrix P, rather than to compute it directly. In this way we can build up very complex matrices, and guarantee that our strategy is locally optimal. This will turn out to be exactly what we need.[10]

To motivate this approach, consider the sybil attack described in Section 3.1. In this attack, there is a large set of dishonest users who report positive interactions with a single dishonest merchant. The result is that total welfare is higher if everyone interacts with the dishonest merchant, rather than no one interacting with them.

But we could improve total welfare further by refining this filtering policy, namely by instructing the honest users to *stop* interacting with the dishonest merchant.

In fact, as long as the users in H are underperforming our benchmark, there is a very simple adjustment that would increase total welfare: tell the users in H to *start* interacting with the resources in S, and *stop* interacting with resources not in S.

So before each decision, we will use matrix prediction to find two matrices, J and C in $\mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$. We then increase P_{xy} when $J_{xy} > 0$, and decrease P_{xy} when $C_{xy} > 0$.

10. Local optimality is actually more useful than global optimality for our purposes. For a static optimization problem global optimality would imply local optimality. But for a learning algorithm that uses a different solution in every round, *competing* with the global optimum does not imply actually converging to the global optimum, and it does not even rule out the existence of a simple local change that would consistently improve performance.

We show that as long as any set of users H is underperforming our benchmark, there is some change that would improve total welfare—namely, we can take $J_{xy} = 1$ for $x \in H, y \in S$ and $C_{xy} = 1$ for $x \in H, y \notin S$. If the users in H are *significantly* underperforming their benchmark, then this change would *significantly* increase total welfare, and so will be eventually discovered by the learning algorithm.

4. The core algorithm

Our algorithms for the collaborative filtering and reputation system problems are both based on a core algorithm $\mathcal{A}$. In this section we define $\mathcal{A}$ and sketch its analysis; the complete analysis is given in Appendix B.

$\mathcal{A}$ solves the special case of collaborative filtering where $\mathcal{Y} = \mathcal{X}$. In this special case there is a single set $\mathcal{X}$ containing both users and resources—for simplicity we will still refer to the elements of $\mathcal{X}$ as "users." An interaction may involve two users $x_0^t, x_1^t \in \mathcal{X}$, but still results in a single payoff p^t.

We can define a number of performance measurements and benchmarks in this model. We recover collaborative filtering by defining $p^{\leq T}(H)$ is defined as $\sum_{t \leq T : x_0^t \in H} p^t$, and the benchmark $\text{OPT}^{\leq T}(H)$ is defined as $\max_{S \subset \mathcal{X}} \sum_{t \leq T : x_0^t \in H, x_1^t \in S} p^t$.

The same algorithm will also be directly applicable to reputation systems in the special case where $p_0^t = p_1^t$.[11] In Section 5 we show how to extend this approach to the case where $p_0^t \neq p_1^t$ (though we still make the "ex ante" symmetry assumption from Section 2.2).

4.1. Online local learning

We first describe the matrix prediction algorithm that will be used to learn the matrices J and C (following the plan described in Section 3.3). To learn these matrices, we will embed them as part of a larger PSD matrix X.

Let $\mathcal{L} = \{\mathbf{j}, \mathbf{c_0}, \mathbf{c_1}\}$ be a set consisting of three possible labels for an element $x \in \mathcal{X}$. Intuitively, in each round we will use online learning to find a labeling $\ell : \mathcal{X} \to \mathcal{L}$. We will tell two users to start interacting if they are both assigned the label $\mathbf{j}$, and we will tell two users to stop interacting if one is assigned the label $\mathbf{c_0}$ and the other is assigned the label $\mathbf{c_1}$.

Formally, we'll embed J and C in a matrix X whose rows and columns are indexed by elements of $\mathcal{X} \times \mathcal{L}$. Intuitively, X represents a probability distribution over possible labelings $\ell : \mathcal{X} \to \mathcal{L}$, via:

$$X_{(x,a)(y,b)} = \mathbb{P}\big(\ell(x) = a \text{ and } \ell(y) = b\big).$$

If X actually corresponds to such a labeling, then X should have all of its entries in $[0, 1]$ and should be positive semi-definite.

We will define J and C as appropriate submatrices of X, via:

$$J_{xy} = X_{(x,\mathbf{j})(y,\mathbf{j})}$$
$$C_{xy} = X_{(x,\mathbf{c_0})(y,\mathbf{c_1})}$$

11. This is not true in general—we present an algorithm which successfully solves both problems, but the two problems are not equivalent.

The interpretation is that two users x_0^t, x_1^t should become more likely to interact if they are both assigned the label $\mathbf{j}$, and they should become less likely to interact if x_0^t is assigned the label $\mathbf{c_0}$ while x_1^t is assigned the label $\mathbf{c_1}$.

In order to learn the matrix X, and hence the matrices J and C, we apply the algorithm $\mathcal{OLL}_\varepsilon$ for online local learning from (Christiano (2014)).

We will be producing a sequence of matrices $X^t \in \mathbb{R}^{(\mathcal{X}\times\mathcal{L})\times(\mathcal{X}\times\mathcal{L})}$, representing a sequence of adjustments to the current filtering policy. We'll represent the payoffs of possible choices by a sequence of matrices $E^t \in \mathbb{R}^{(\mathcal{X}\times\mathcal{L})\times(\mathcal{X}\times\mathcal{L})}$: the payoff of choosing the matrix $X^t \in \mathbb{R}^{(\mathcal{X}\times\mathcal{L})\times(\mathcal{X}\times\mathcal{L})}$ is given by the matrix inner product

$$\left\langle E^t, X^t \right\rangle = \sum_{x,y\in\mathcal{X},a,b\in\mathcal{L}} E^t_{(x,a)(y,b)} X^t_{(x,a)(y,b)}.$$

Intuitively, $E^t_{(x,a)(y,b)}$ represents a payoff received by any labeling ℓ with $\ell(x) = a$ and $\ell(y) = b$. In our application the matrices E will be very sparse.

Write $E^{\leq T} = \sum_{t\leq T} E^t$ and $E^{<T} = \sum_{t<T} E^t$.

Then $\mathcal{OLL}_\varepsilon\left(E^{<T}\right)$ is defined as the solution to the following semidefinite program:[12]

$$\begin{array}{c}\operatorname{argmax} \\ X\in\mathbb{R}^{(\mathcal{X}\times\mathcal{L})\times(\mathcal{X}\times\mathcal{L})} \\ X\succeq 0 \\ 0\leq X_{(x,a)(y,b)}\leq 1\end{array} \varepsilon \left\langle E^{<T}, X \right\rangle + \log\det(X+I).$$

The most important property of $\mathcal{OLL}_\varepsilon$ is given by the following result, which states that the payoffs of using $\mathcal{OLL}_\varepsilon\left(E^{<t}\right)$ are nearly as good as using the best fixed PSD matrix X:

Theorem *[Restatement of Theorem 12] For any sequence E^t and any $X \succeq 0$ with entries in $[0,1]$, we have*

$$\sum_{t<T} \left\langle E^t, \mathcal{OLL}_\varepsilon\left(E^{<t}\right)\right\rangle \geq \sum_{t<T} \left\langle E^t, X\right\rangle - \mathcal{O}\left(\varepsilon \sum_{t<T} \left|E^t\right|_1^2\right) - \mathcal{O}\left(N\varepsilon^{-1}\right)$$

The analysis will also use the fact that the matrices X^t change slowly:

Lemma *[Restatement of Lemma 11] For any $E^{<t}$ and any E^t we have*

$$\left|\mathcal{OLL}_\varepsilon\left(E^{<t}\right) - \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)\right|_\infty \leq \mathcal{O}\left(\varepsilon \left|E^t\right|_1\right)$$

Our statement of Theorem 12 and Lemma 11 are slightly different from the statements in (Christiano (2014)), so we present proofs in Appendix E.

12. The algorithm in (Christiano (2014)) maximized under some additional equality constraints, e.g. $\sum_a X_{(x,a)(x,a)} = 1$. For our purposes it doesn't matter whether we include these or not, and for ease of exposition we omit them. It is easy to verify that this leaves the results of (Christiano (2014)) unchanged, as we show in Appendix E; likewise, including these extra inequalities would leave our results unchanged.

4.2. Our core algorithm

We will define an algorithm $\mathcal{A}_\varepsilon$ which is parameterized by a learning rate $\varepsilon < \frac{1}{4}$. In order to implement the unparameterized version we will use a standard doubling trick, restarting $\mathcal{A}_\varepsilon$ with exponentially decreasing values of ε.

The algorithm $\mathcal{A}_\varepsilon$:

- Set $P^0_{xy} = 1$ for $x, y \in \mathcal{X}$.
- For $t = 0, 1, \ldots, T-1$:
  - Output $s^t = 1$ with probability $P^t_{x_0^t x_1^t}$, and $s^t = 0$ otherwise.
  - If $s^t = 1$ set $\hat{p}^t = p^t / P^t_{x_0^t x_1^t}$. Otherwise, set $\hat{p}^t = 0$.
  - Define $E^t \in \mathbb{R}^{(\mathcal{X}\times\mathcal{L})\times(\mathcal{X}\times\mathcal{L})}$ to be the sparse matrix with the following non-zero entries:

$$E^t_{(x_0^t,\mathbf{j})(x_1^t,\mathbf{j})} = (1 - P^t_{x_0^t x_1^t})\hat{p}^t$$
$$E^t_{(x_0^t,\mathbf{c_0})(x_1^t,\mathbf{c_1}} = -P^t_{x_0^t x_1^t}\hat{p}^t$$

  - Let $X^{t+1} = \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)$.
  - Define

$$J^{t+1}_{xy} = \frac{1}{4} X^{t+1}_{(x,\mathbf{j})(y,\mathbf{j})} + \varepsilon^{1/2}$$
$$C^{t+1}_{xy} = \frac{1}{4} X^{t+1}_{(x,\mathbf{c_0})(y,\mathbf{c_1})}$$

  - For each x, y, set

$$P^{t+1}_{xy} = P^t_{xy} - C^{t+1}_{xy} P^t_{xy} + J^{t+1}_{xy}(1 - P^t_{xy}).$$

4.3. Analysis sketch

Our matrix prediction algorithm ensures that

$$\sum P^t_{x^t y^t} p^t \geq \sum \left(P^{t-1}_{x^t y^t} p^t - C^*_{xy} P^{t-1}_{x^t y^t} p^t + J^*_{x^t y^t} \left(1 - P^{t-1}_{x^t y^t}\right) p^t \right) - R(T)$$

for any matrices C^* and J^* in the comparison class, where $R(T)$ is a regret bound whose value depends on the learning rate.

In particular, we can pick C^* and J^* to be:

$$C^*_{xy} = \begin{cases} 1 & \text{if } x \in H \text{ and } y \notin S \\ 0 & \text{otherwise} \end{cases} \quad J^*_{xy} = \begin{cases} 1 & \text{if } x \in H \text{ and } y \in S \\ 0 & \text{otherwise} \end{cases}$$

This implies that

$$\sum P^t_{x^t y^t} p^t \geq \left(\sum_t \begin{cases} p^t & \text{if } x^t \in H, y^t \in S \\ 0 & \text{if } x^t \in H, y^t \notin S \\ P^{t-1}_{x^t y^t} p^t & \text{otherwise} \end{cases} \right) - R(T) \tag{1}$$

$$= \sum P^{t-1}_{x^t y^t} p^t + \text{OPT}^{\leq T}(H) - p^{\leq T}(H) - R(T) \tag{2}$$

If the P^{t-1} on the right hand side of Equation 1 were replaced by a P^t, then this would yield precisely the desired result.

To finish our argument, we use the fact that our matrix prediction algorithm guarantees that C and J change slowly. By imposing a lower bound on the entries of the matrix J, we can ensure that P also changes slowly. This lets us replace P^{t-1} by P^t, and yields the desired result. Unfortunately, the replacement slightly weakens the bound, which leads to suboptimal $T^{2/3}$ regret.

The complete analysis is given in Appendix B

5. Reputation systems with asymmetrical payoffs

In the reputation system case, if payoffs are symmetric (with $p^t_0 = p^t_1$) then we could apply the algorithm from Section 4. But the asymmetric case presents new difficulties. In this section we sketch our solution. A formal algorithm and analysis is given in Appendix D.

Our approach is to reduce to the symmetric case by implementing "side payments" that track a user's debt to the rest of the community. We maintain an account balance $w^t(x) > 0$ for each user x. Whenever $p^t_0 \neq p^t_1$, we adjust $w\left(x^t_0\right)$ and $w\left(x^t_1\right)$ so that the "effective payoffs" of the two users are equal and the total balance $\sum_{x \in \mathcal{X}} w^t(x)$ is conserved. Over the long-run the budget constraints $w^t(x) \geq 0$ imply that the actual payoffs cannot be too different from the effective payoffs.

The most natural approach is to consider changes in balance as interchangeable with payoffs. That is, we could pick the new balances w^{t+1} such that the two users benefit equally:

$$p^t_0 + w^{t+1}\left(x^t_0\right) - w^t\left(x^t_0\right) = p^t_1 + w^{t+1}\left(x^t_1\right) - w^t\left(x^t_1\right).$$

The problem with this approach is that we can't ensure that all balances $w^{t+1}(x)$ remain positive. If we restrict interactions with user x when $w(x)$ is low, then once a user goes to a low value they will simply get "stuck," unable to recover. And if we allow users to run up arbitrarily negative balances, then a dishonest user may do arbitrarily much damage.

Motivated by the idea of logarithmic potentials in optimization, we instead define the effective payoff as the actual payoff plus the change in *log* balance. That is, we pick the new balances w^{t+1} such that

$$p^t_0 + \log\left(w^{t+1}\left(x^t_0\right)\right) - \log\left(w^t\left(x^t_0\right)\right) = p^t_1 + \log\left(w^{t+1}\left(x^t_1\right)\right) - \log\left(w^t\left(x^t_1\right)\right).$$

The logarithmic potential function prevents $w^t(x)$ from ever getting too close to 0 without introducing significant distortions. The resulting protocol seems to be self-correcting and highly robust to liquidity shortages. It also has a natural interpretation in terms of logarithmic utility functions, and may be of independent interest.

More formally, in round t we define the transfer

$$\tau^t \propto \frac{w_0^t w_1^t \left(p_1^t - p_0^t\right)}{\left(w_0^t + w_1^t\right)},$$

where $w_i^t = w^t\left(x_i^t\right)$. We then set

$$w^{t+1}\left(x_0^t\right) = w^t\left(x_0^t\right) - \tau^t$$
$$w^{t+1}\left(x_1^t\right) = w^t\left(x_1^t\right) + \tau^t$$

We define the symmetric payoff of the interaction as the quantity

$$p^t = p_i^t + \Delta \log\left(w\left(x_i^t\right)\right)$$

which is approximately equal for $i = 0$ and $i = 1$. Moreover, it is easy to check that the expectation of p^t is at least the minimum of the expectation of p_0^t and the expectation of p_1^t, which is precisely what we need in order to compete with $\text{OPT}_{\text{sym}}^{\leq T}(H)$.

We can now apply $\mathcal{A}$ to the symmetric case. The analysis of $\mathcal{A}$ case guarantees that total payoff p^t of honest users is at least $\text{OPT}_{\text{sym}}^{\leq T}(H)$. So whenever the actual payoff of the honest users is below $\text{OPT}_{\text{sym}}^{\leq T}(H)$, it must be because their total balance $w^t(x)$ is increasing.

Because $w^t(x) > 0$, the *total* weight of honest users, $\sum_{x \in H} w^t(x)$, is bounded by N. So we can obtain a bound on the actual payoff—the worst case is when users in H end up with *all* of the weight (when they are owed the maximal possible debt by users outside of H). But even in this case the gap between the symmetric payoffs and the real payoffs is bounded by $|H| \log\left(\frac{|H|}{N}\right)$.

6. Conclusion

Our results apply in a simple stylized model of reputation systems and collaborative filtering, and they could be improved in many directions; some of these are explored in Appendix F. But we have shown that there is no in-principle obstruction to building reputation systems which are statistically efficient and theoretically secure. We hope that this positive result may encourage other researchers to develop improved protocols and apply them in more complex domains.

References

Alon, Awerbuch, Azar, and Patt-Shamir. Tell me who I am: An interactive recommendation system. In *SPAA: Annual ACM Symposium on Parallel Algorithms and Architectures*, 2006.

Awerbuch and Kleinberg. Competitive collaborative learning. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 2005.

Awerbuch, Azar, Lotker, Patt-Shamir, and Tuttle. Collaborate with strangers to find own preferences. *MST: Mathematical Systems Theory*, 42, 2008.

Paul Christiano. Online local learning via semidefinite programming. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2014.

B. Cohen. Incentives build robustness in bittorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, 2003. URL http://citeseer.nj.nec.com/cohen03incentives.html.

Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the 13th International Workshop on Network and Operating System Support for Digital Audio and Video Archive (NOSSDAV-03)*, pages 144–152, New York, June 1–3 2003. ACM Press.

Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *CoRR*, abs/1204.0136, 2012. URL http://arxiv.org/abs/1204.0136.

Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In Gusztáv Hencsey, Bebo White, Yih-Farn Robin Chen, László Kovács, and Steve Lawrence, editors, *WWW*, pages 640–651. ACM, 2003. ISBN 1-58113-680-3. URL http://doi.acm.org/10.1145/775152.775242.

M. Meulpolder, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. Bartercast: A practical approach to prevent lazy freeriding in P2P networks. In State University of New York Yuanyuan Yang, editor, *Proceedings of the 23rd IEEE International Parallel & Distributed Processing Symposium*, pages 1–8, Los Alamitos, USA, May 2009. IEEE Computer Society. ISBN 978-1-4244-3750-4. URL http://www.pds.ewi.tudelft.nl/%7Eepema/Papers/2009/HotP2P2009.pdf.

Paul Resnick and Rahul Sami. The influence limiter: provably manipulation-resistant recommender systems. In Joseph A. Konstan, John Riedl, and Barry Smyth, editors, *RecSys*, pages 25–32. ACM, 2007. ISBN 978-1-59593-730-8. URL http://doi.acm.org/10.1145/1297231.1297236.

Sven Seuken and David C. Parkes. Sybil-proof accounting mechanisms with transitive trust. In Ana L. C. Bazzan, Michael N. Huhns, Alessio Lomuscio, and Paul Scerri, editors, *AAMAS*, pages 205–212. IFAAMAS/ACM, 2014. ISBN 978-1-4503-2738-1. URL http://dl.acm.org/citation.cfm?id=2615731.

Sven Seuken, Jie Tang, and David C. Parkes. Accounting mechanisms for distributed work systems. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1802.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. URL http://dx.doi.org/10.1561/2200000018.

Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. Artificial Intellegence*, 2009, 2009. URL http://dx.doi.org/10.1155/2009/421425.

Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gun Sirer. KARMA : A secure economic framework for peer-to-peer resource sharing, November 13 2003. URL http://citeseer.ist.psu.edu/682464.html;http://www.cs.cornell.edu/People/egs/papers/karma.pdf.

Haifeng Yu, Chenwei Shi, Michael Kaminsky, Phillip B. Gibbons, and Feng Xiao. DSybil: Optimal sybil-resistance for recommendation systems. In *IEEE Symposium on Security and Privacy*, pages 283–298. IEEE Computer Society, 2009. ISBN 978-0-7695-3633-0. URL http://dx.doi.org/10.1109/SP.2009.26.

Appendix A. A unified model

In this section we introduce a unified model for collaborative filtering and reputation systems with symmetric payoffs. This model is essentially the special case of our collaborative filtering model from Section 2.1 with $\mathcal{Y} = \mathcal{X}$, though we also define some new performance measures that will be used in Appendix B.

The algorithm makes decisions over a series of rounds $t = 1, 2, \ldots$. Prior to the first round, the algorithm is given a set $\mathcal{X}$[13] of size $N = |\mathcal{X}|$. The set $\mathcal{X}$ contains all users of the system as well as all resources with which those users might interact: in the setting of Section 2.1, $\mathcal{X}$ plays the role of $\mathcal{X} \cup \mathcal{Y}$.

In each round:

1. An adversary chooses $x_0^t, x_1^t \in \mathcal{X}$ and reveals them to the algorithm.
2. An adversary chooses $p^t \in [-1, 1]$, but does not reveal it to the algorithm.
3. The algorithm outputs $s^t \in \{0, 1\}$.
4. If $s^t = 1$, then the algorithm observes p^t.

By convention, we assume that in the collaborative filtering setting, x_1^t is the resource and receives a payoff $p_1^t = 0$.

13. In fact our algorithm and analysis would apply even if users join and leave over the course of the algorithm. $\mathcal{X}$ would then be a function of T, containing all users who participate in any of the rounds $t = 1, 2, \ldots, T$. For simplicity, we assume that $\mathcal{X}$ is fixed.

For any set $H \subset \mathcal{X}$, we can define three different performance measures:

$$p_0^{\leq T}(H) = \sum_{t \leq T : x_0^t \in H} s^t p^t$$
$$p_1^{\leq T}(H) = \sum_{t \leq T : x_1^t \in H} s^t p^t$$
$$p^{\leq T}(H) = p_0^{\leq T}(H) + p_1^{\leq T}(H)$$

All of these measures use the same benchmark:

$$\text{OPT}^{\leq T}(H) = \sum_{t \leq T : x_0^t, x_1^t \in H} p^t$$

where the expectations are taken with respect to the distributions chosen by nature in step 2.

Appendix B. Analyzing $\mathcal{A}$

The algorithm $\mathcal{A}$ is defined in Section 4. In this section we prove:

Theorem 3 *For any set of users H, any $T > 0$, and each $i \in \{0, 1\}$, $\mathcal{A}$ satisfies*

$$p_i^{\leq T}(H) \geq \mathit{OPT}^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right).$$

Because the same algorithm satisfies the guarantee for both $i = 0$ and $i = 1$, we can add the corresponding inequalities to prove $p^{\leq T}(H) \geq 2\text{OPT}^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right)$.

In the rest of this section we prove Theorem 3. This proof follows the informal exposition in Section 4.3.

In Appendix C we apply $\mathcal{A}$ to the collaborative filtering case, defining the algorithm $\mathcal{A}_{\text{filtering}}$ and proving Theorem 1. In Appendix D we apply $\mathcal{A}$ to the reputation system case, defining the algorithm $\mathcal{A}_{\text{reputation}}$ and proving Theorem 2

For convenience, we will write $p^t(P)$ for the expected payoff obtained by using the matrix P in round t, namely

$$p^t(P) = p^t P_{x_0^t x_1^t}.$$

We write $U\left(P^t, X\right)$ for the matrix that would be obtained by updating P^t with the matrix $X^{t+1} = X$, i.e. defined by the equations

$$J_{xy} = \frac{1}{4} X_{(x,\mathbf{j})(y,\mathbf{j})} + \varepsilon^{1/2}$$
$$C_{xy} = \frac{1}{4} X_{(x,\mathbf{c_0})(y,\mathbf{c_1})}$$
$$U\left(P^t, X\right)_{xy} = P^t_{xy} - C_{xy} P^t_{xy} + J_{xy}\left(1 - P^t_{xy}\right)$$

B.1. Performance lemmas

Our first lemma asserts that using the matrices X^t to update P is nearly as good as using any fixed PSD matrix X. This is what allows us to argue that our algorithm will find a good update if one exists: with this lemma in hand, we simply need to show that the desired update is implemented by some PSD matrix X.

Lemma 4 *For any $X \succeq 0$ with $0 \leq X_{(x,a)(y,b)} \leq 1$ we have*

$$\sum_{t \leq T} p^t\Big(P^t\Big) \geq \sum_{t \leq T} p^t\Big(U\Big(P^{t-1}, X\Big)\Big) - \mathcal{O}\Big(\varepsilon \sum \Big|\hat{p}^t\Big|^2\Big) - \mathcal{O}\Big(\frac{N}{\varepsilon}\Big),$$

where P^t are the matrices computed by $\mathcal{A}_\varepsilon$.

Proof We have $p^t\big(P^t\big) = p^t\Big(U\big(P^{t-1}, X^t\big)\Big)$. Moreover,

$$\begin{aligned} p^t\Big(U\Big(P^{t-1}, X\Big)\Big) &= p^t\Big((1 - J_{x_0^t x_1^t} - C_{x_0^t x_1^t})P^{t-1}_{x_0^t x_1^t} + J_{x_0^t x_1^t}\Big) \\ &= p^t\Big(P^{t-1}_{x_0^t x_1^t}\Big) + J_{x_0^t x_1^t} p^t\Big(1 - P^{t-1}_{x_0^t x_1^t}\Big) - C_{x_0^t x_1^t} p^t P^{t-1}_{x_0^t x_1^t} \\ &= p^t\Big(P^{t-1}\Big) + \frac{1}{4}\Big\langle X, E^t\Big\rangle + \beta^t, \end{aligned}$$

where J and C are defined from X in the same way that J^t and C^t are defined from X^t, and $\beta^t = \varepsilon^{1/2} p^t\Big(1 - P^{t-1}_{x_0^t x_0^t}\Big)$ doesn't depend on X. We constructed the matrix E^t precisely so that it would satisfy this identity.

Because $X^t = \mathcal{OLL}_\varepsilon\big(E^{<t}\big)$, by Theorem 12 we have

$$\begin{aligned} \sum_{t \leq T} p^t\Big(P^t\Big) &= \sum_{t \leq T}\Big(p^t\Big(P^{t-1}\Big) + \frac{1}{4}\Big\langle X^t, E^t\Big\rangle + \beta^t\Big) \\ &\geq \sum_{t \leq T}\Big(p^t\Big(P^{t-1}\Big) + \frac{1}{4}\Big\langle X, E^t\Big\rangle + \beta^t\Big) - \mathcal{O}\Big(\varepsilon \sum \Big|E^t\Big|_1^2\Big) - \mathcal{O}\Big(\frac{N}{\varepsilon}\Big) \\ &= \sum_{t \leq T} p^t\Big(U\Big(P^{t-1}, X\Big)\Big) - \mathcal{O}\Big(\varepsilon \sum \Big|E^t\Big|_1^2\Big) - \mathcal{O}\Big(\frac{N}{\varepsilon}\Big) \\ &= \sum_{t \leq T} p^t\Big(U\Big(P^{t-1}, X\Big)\Big) - \mathcal{O}\Big(\varepsilon \sum \Big|\hat{p}^t\Big|^2\Big) - \mathcal{O}\Big(\frac{N}{\varepsilon}\Big), \end{aligned}$$

where the last equality holds because $\big|E^t\big|_1 = \mathcal{O}\big(\big|\hat{p}^t\big|\big)$. ■

The next step is to prove that *if* some group H of users is not meeting our benchmark, *then* there is some PSD recommendation which would generate a substantial improvement in total payoff.

This essentially corresponds to the description of the matrices J^*, C^* in the proof sketch in Section 4.3.

The reason for the $\varepsilon^{1/2}$ loss in this theorem is our restriction that $J^t \geq \varepsilon^{1/2}$. This limits our ability to stop honest and dishonest users from interacting, but was needed to prevent rapid changes in P^t.

Lemma 5 *For any set H there is a matrix $X^H \succeq 0$, with $0 \leq X^H_{(x,a)(y,b)} \leq 1$, such that*

$$OPT_i^{\leq T}(H) - p_i^{\leq T}(H) = \mathcal{O}\left(\sum_{t \leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right) + \varepsilon^{1/2}\left|\hat{p}^t\right|\right)\right),$$

where P^t are the matrices produced by $\mathcal{A}_\varepsilon$.

Proof Observe that $\mathcal{A}_\varepsilon$ is perfectly symmetrical under an exchange of x_0^t and x_1^t: the only difference is that the matrices P^t are transposed, and the indices of $X_{(x,a)(y,b)}$ are modified by exchanging $\mathbf{c_0}0$ and $\mathbf{c_1}$. So it suffices to prove the theorem for $i = 0$.

Define the vector v^H via $v_{(x,\mathbf{j})} = v_{(x,\mathbf{c_0})} = 1$ for $x \in H$, $v_{(x,\mathbf{c_1})} = 1$ for $x \notin H$, and $v_{(x,a)} = 0$ otherwise. Define the matrix X^H as $\left(v^H\right)\left(v^H\right)^T$. This matrix is clearly PSD, and has entries in $[0, 1]$.

Suppose $x \notin H$. Then $X^H_{(x,\mathbf{c_0})(y,\mathbf{c_1})} = X^H_{(x,\mathbf{j})(y,\mathbf{j})} = 0$ for all y. Thus $U\left(P^t, X^H\right)_{xy} = P^t_{xy}$. For $x, y \in H$, we have $X^H_{(x,\mathbf{j})(y,\mathbf{j})} = 1$ and hence

$$U\left(P^t, X^H\right)_{xy} = \frac{3}{4}P^t_{xy} + \frac{1}{4} + \varepsilon^{1/2}(1 - P^t_{xy}).$$

For $x \in H, y \notin H$, we have $X^H_{(x,\mathbf{c_0})(y,\mathbf{c_1})} = 1$ and hence

$$U\left(P^t, X^H\right)_{xy} = \frac{3}{4}P^t_{xy} + \varepsilon^{1/2}(1 - P^t_{xy}).$$

We compute:

$$\begin{aligned}\sum_{t \leq T : x_0^t \in H} p^t\left(U\left(P^t, X^H\right)\right) &\geq \frac{3}{4}\sum_{t \leq T : x_0^t \in H} p^t\left(P^t\right) + \frac{1}{4}\sum_{t \leq T : x_0^t, x_1^t \in H} p^t - \varepsilon^{1/2}\sum\left|p^t\right| \\ &= p_i^{\leq T}(H) - \frac{1}{4}p_i^{\leq T}(H) + \frac{1}{4}\text{OPT}_i^{\leq T}(H) - \varepsilon^{1/2}\sum\left|p^t\right|\end{aligned}$$

Rearranging, and using the fact that $U\left(P^t, X^H\right)_{xy} = P^t_{xy}$ for $x \notin H$:

$$\begin{aligned}\text{OPT}_i^{\leq T}(H) - p_i^{\leq T}(H) &\leq 4\sum_{t \leq T : x_0^t \in H}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right) + \varepsilon^{1/2}\left|p^t\right|\right) \\ &= 4\sum_{t \leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right) + \varepsilon^{1/2}\left|p^t\right|\right),\end{aligned}$$

as desired. ■

B.2. Bounding rates of change

The other ingredient of our proof is the claim that P changes slowly. This corresponds to the final paragraph of the proof sketched in Section 4.3, and allows us to replace P^{t-1} with P^t in the conclusion of Lemma 4.

Lemma 6 *For any x, y, we have*

$$\sum_{t<T} \left| P_{xy}^{t+1} - P_{xy}^{t} \right| = \mathcal{O}\left(\varepsilon^{1/2} \sum_{t<T} \left| \hat{p}^t \right| \right) + \mathcal{O}(1),$$

where P^t are the matrices computed by $\mathcal{A}_\varepsilon$.

Proof Consider the "steady state" value of P given J and C, defined as

$$Y_{xy}^t = \frac{J_{xy}^t}{J_{xy}^t + C_{xy}^t}.$$

This is the value to which P would converge if J and C were left unchanged for many iterations. We will show that Y^t does not change very quickly, and that P^t is "following" Y^t and hence does not change very quickly either.

Note that each entry of J^t and C^t is defined to be an affine function of one entry of $\mathcal{OLL}_\varepsilon\left(E^{<t}\right)$ with coefficients that are $\mathcal{O}(1)$. Because $\left|E^t\right|_1$ is $\mathcal{O}\left(\hat{p}^t\right)$, Lemma 11 implies that $\left|J_{xy}^{t+1} - J_{xy}^t\right| = \mathcal{O}\left(\varepsilon \left|\hat{p}^t\right|\right)$, and similarly $\left|C_{xy}^{t+1} - C_{xy}^t\right| = \mathcal{O}\left(\varepsilon \left|\hat{p}^t\right|\right)$. By definition we have $C_{xy}^t \geq 0$ and $1 \geq J_{xy}^t \geq \varepsilon^{1/2}$. Thus

$$\begin{aligned}
\left|Y_{xy}^{t+1} - Y_{xy}^t\right| &= \left| \frac{J_{xy}^{t+1}}{J_{xy}^{t+1} + C_{xy}^{t+1}} - \frac{J_{xy}^t}{J_{xy}^t + C_{xy}^t} \right| \\
&= \left| \frac{J_{xy}^{t+1}\left(J_{xy}^t + C_{xy}^t\right) - J_{xy}^t\left(J_{xy}^{t+1} + C_{xy}^{t+1}\right)}{\left(J_{xy}^t + C_{xy}^t\right)\left(J_{xy}^{t+1} + C_{xy}^{t+1}\right)} \right| \\
&= \left| \frac{J_{xy}^{t+1}\left(C_{xy}^t - C_{xy}^{t+1}\right) + C_{xy}^{t+1}\left(J_{xy}^{t+1} - J_{xy}^t\right)}{J_{xy}^t\left(J_{xy}^{t+1} + C_{xy}^{t+1}\right)} \right| \\
&= \mathcal{O}\left(\frac{\varepsilon\hat{p}^t J_{xy}^{t+1} + \varepsilon\hat{p}^t C_{xy}^{t+1}}{J_{xy}^t\left(J_{xy}^{t+1} + C_{xy}^{t+1}\right)} \right) = \mathcal{O}\left(\frac{\varepsilon\hat{p}^t}{J_{xy}^t} \right) = \mathcal{O}\left(\frac{\varepsilon\hat{p}^t}{\varepsilon^{1/2}} \right) = \mathcal{O}\left(\varepsilon^{1/2}\hat{p}^t \right)
\end{aligned}$$

Note that this conclusion requires $J_{xy}^t \geq \varepsilon^{1/2}$; enforcing this condition is responsible for our sub-optimal regret bounds. Otherwise both J_{xy}^t and C_{xy}^t could take on very small values and P_{xy}^t could change rapidly.

By the definition of P^{t+1}, we have

$$\begin{aligned} P^{t+1}_{xy} &= (1 - J^t_{xy} - C^t_{xy})P^t_{xy} + J^t_{xy} \\ &= (1 - J^t_{xy} - C^t_{xy})P^t_{xy} + (J^t_{xy} + C^t_{xy})Y^t_{xy} \end{aligned}$$

By the definition of J and C, we have $J^t_{xy} + C^t_{xy} \leq \frac{1}{4} + \frac{1}{4} + \varepsilon^{1/2} \leq 1$. Thus P^{t+1} is a convex combination of P^t_{xy} and Y^t_{xy}. In particular,

$$\left|P^t_{xy} - Y^t_{xy}\right| = \left|P^t_{xy} - P^{t+1}_{xy}\right| + \left|P^{t+1}_{xy} - Y^t_{xy}\right|.$$

Intuitively this suggests that $\sum \left|P^{t+1}_{xy} - P^t_{xy}\right| \leq \sum \left|Y^{t+1}_{xy} - Y^t_{xy}\right|$, from which the desired result would follow.

To see this formally, consider the total length of the path

$$P^0_{xy}, P^1_{xy}, \ldots, P^t_{xy}, Y^t_{xy},$$

i.e. the sum of the absolute values of the differences between consecutive points. If we change this path by adding a new point:

$$P^0_{xy}, P^1_{xy}, \ldots, P^t_{xy}, P^{t+1}_{xy}, Y^t_{xy}$$

we don't change the total length, because P^{t+1} is in between P^t and Y^t. If we then change this path by changing the last point:

$$P^0_{xy}, P^1_{xy}, \ldots, P^t_{xy}, P^{t+1}_{xy}, Y^{t+1}_{xy}$$

we increase the total length by at most $\left|Y^{t+1}_{xy} - Y^t_{xy}\right|$.

By induction, the sum $\sum \left|P^{t+1}_{xy} - P^t_{xy}\right|$ is at most $\sum \left|Y^{t+1}_{xy} - Y^t_{xy}\right| + \mathcal{O}(1) = \mathcal{O}\left(\varepsilon^{1/2} \sum \left|\hat{p}^t\right|\right) + \mathcal{O}(1)$. ■

B.3. Putting it all together

We are now ready to prove the main result of this section:

Theorem 7 *For any set of users H and any $T > 0$, $\mathcal{A}_\varepsilon$ satisfies*

$$p_i^{\leq T}(H) \geq \mathit{OPT}^{\leq T}(H) - \mathcal{O}\left(\varepsilon^{1/2} T\right) - \mathcal{O}\left(\frac{N}{\varepsilon}\right)$$

in expectation.

Proof By Lemma 5, there exists an $X^H \succeq 0$ with $0 \leq X^H_{(a,i)(b,j)} \leq 1$ such that

$$\begin{aligned} \mathrm{OPT}_i^{\leq T}(H) - p_i^{\leq T}(H) &= \mathcal{O}\left(\sum_{t \leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right) + \varepsilon^{1/2}\left|\hat{p}^t\right|\right)\right) \\ &\leq \mathcal{O}\left(\sum_{t \leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right)\right) + \varepsilon^{1/2} T\right) \end{aligned}$$

in expectation, where the second inequality holds because $\mathbb{E}\left[\left|\hat{p}^t\right|\right] \leq 1$. So it suffices to bound $\sum_{t\leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t(P^t)\right)$.

Note that $\mathcal{A}_\varepsilon$ always satisfies $P^t_{x_0^t x_1^t} \geq J^t_{x_0^t x_1^t} \geq \varepsilon^{1/2}$, and consequently $\left|\hat{p}^t\right| \leq 1/P^t_{x_0^t x_1^t} = \varepsilon^{-1/2}$. We have in expectation:

$$
\begin{aligned}
&\sum_{t\leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(P^t\right)\right)\\
&= \sum_{t\leq T}\left(p^t\left(U\left(P^t, X^H\right)\right) - p^t\left(U\left(P^{t-1}, X^H\right)\right)\right) + \sum_{t\leq T}\left(p^t\left(U\left(P^{t-1}, X^H\right)\right) - p^t\left(P^t\right)\right)\\
&\leq \sum_{t\leq T}\left|p^t\right|\left|U\left(P^t, X^H\right) - U\left(P^{t-1}, X^H\right)\right|_\infty + \mathcal{O}\left(\varepsilon\sum\left|\hat{p}^t\right|^2\right) + \mathcal{O}\left(\frac{N}{\varepsilon}\right)\\
&\leq \sum_{t\leq T}\left|p^t\right|\left|P^t - P^{t-1}\right|_\infty + \mathcal{O}\left(\varepsilon^{1/2}\sum\left|p^t\right|\right) + \mathcal{O}\left(\frac{N}{\varepsilon}\right)\\
&\leq \sum_{t\leq T}\left|P^t - P^{t-1}\right|_\infty + \mathcal{O}\left(\varepsilon^{1/2}\sum\left|\hat{p}^t\right|\right) + \mathcal{O}\left(\frac{N}{\varepsilon}\right)\\
&\leq \varepsilon^{1/2}\sum_{t\leq T}\left|p^t\right| + \mathcal{O}\left(\varepsilon^{1/2}\sum\left|p^t\right|\right) + \mathcal{O}\left(\frac{N}{\varepsilon}\right)\\
&= \mathcal{O}\left(\varepsilon^{1/2}T\right) + \mathcal{O}\left(\frac{N}{\varepsilon}\right)
\end{aligned}
$$

where the third line uses the definition of p^t and Theorem 4, the fourth line uses the definition of $U(\cdot,\cdot)$ and the upper bound $\left|\hat{p}^t\right| \leq \varepsilon^{-1/2}$, the fifth line uses the bound $\mathbb{E}\left[\left|\hat{p}^t\right|\right] \leq 1$ (note that P^t is fixed before p^t is sampled), the sixth line uses Theorem 6, and the seventh line uses the bound $\mathbb{E}\left[\left|\hat{p}^t\right|\right] \leq 1$.

■

By setting $\varepsilon = \frac{N^{2/3}}{T^{2/3}}$ and using a standard doubling trick, we obtain a more convenient form:

Theorem *[Restatement of Theorem 3] There is an algorithm $\mathcal{A}$ which satisfies*

$$p_i^{\leq T}(H) \geq \mathit{OPT}^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right).$$

Proof Every time $T = 2^k$, we start a new copy of $\mathcal{A}_\varepsilon$ with $\varepsilon = \varepsilon_k = N^{2/3}T^{-2/3}$.

By Theorem 7, we have

$$
\begin{aligned}
\sum_{\substack{2^k\leq t<2^{k+1}\\ x_i t\in H}} p^t s^t &\geq \sum_{\substack{2^k\leq t<2^{k+1}\\ x_0^t, x_1^t\in H}} p^t - \mathcal{O}\left(\varepsilon_k^{1/2}2^k\right) - \mathcal{O}\left(\frac{N}{\varepsilon_k}\right)\\
&= \sum_{\substack{2^k\leq t<2^{k+1}\\ x_0^t, x_1^t\in H}} p^t - \mathcal{O}\left(2^{2k/3}N^{1/3}\right)
\end{aligned}
$$

Summing from $k = 0$ to $k = \lceil \log(T) \rceil$, we obtain the desired bound. ■

Appendix C. Algorithm for collaborative filtering

In this section we define the algorithm $\mathcal{A}_{\text{filtering}}$. It is a direct application of the core algorithm $\mathcal{A}$, which was defined in Section 4 and analyzed in Section B.

We prove

Theorem *[Restatement of Theorem 1] For any set of users $H \subset \mathcal{X}$ and any $T > 0$, the recommendations of $\mathcal{A}_{filtering}$ satisfy*

$$p^{\leq T}(H) \geq OPT^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right)$$

where $N = |\mathcal{X} \cup \mathcal{Y}|$.

Proof Given $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{A}_{\text{filtering}}$ initializes a copy of $\mathcal{A}$ with set of users $\mathcal{X} \cup \mathcal{Y}$. In each round $t = 1, 2, \ldots$, $\mathcal{A}_{\text{filtering}}$ consults $\mathcal{A}$ to determine if $x_0^t = x^t$ and $x_1^t = y^t$ should interact. It then passes the payoff p^t to $\mathcal{A}$.

We apply Theorem 3 with $i = 0$ and with H containing both the users we care about, and all of the resources from the maximizing set S. ■

Appendix D. Algorithm for reputation systems

In this section we define an algorithm $\mathcal{A}_{\text{reputation}}$ for the reputation system setting, and prove

Theorem *[Restatement of Theorem 2] For any set of users H and any $T > 0$, $\mathcal{A}_{reputation}$ satisfies*

$$p^{\leq T}(H) \geq OPT^{\leq T}_{sym}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right).$$

The algorithm $\mathcal{A}_{\text{reputation}}$ works as follows:

- Given $\mathcal{X}$, initialize an instance of $\mathcal{A}$ with set of users $\mathcal{X}$.
- For each $x \in \mathcal{X}$, set $w^0(x) = 1$.
- For each round $t = 1, 2, \ldots$
  - Consult $\mathcal{A}$ to decide whether x_0^t and x_1^t should interact. Output the recommendation of $\mathcal{A}$.
  - If no interaction occurs, go on to the next round.
  - If interaction occurs, define $w_i^t = w^t(x_i^t)$ and $\beta_i^t = \frac{w_i^t}{w_0^t + w_1^t}$. Set $p^t = \beta_0^t p_0^t + \beta_1^t p_1^t$ and return this payoff to the $\mathcal{A}$ as the payoff of the interaction.

– Set

$$\tau^t = \frac{\delta w_0^t w_1^t (p_1^t - p_0^t)}{(w_0^t + w_1^t)},$$

where $\delta < \frac{1}{4}$ is a learning rate parameter whose value will be defined later.

– Set

$$w^{t+1}\left(x_0^t\right) = w^t\left(x_0^t\right) + \tau^t,$$
$$w^{t+1}\left(x_1^t\right) = w^t\left(x_1^t\right) - \tau^t,$$

and go on to the next round.

For convenience, write $\tau_i^t = (-1)^i \tau^t$, i.e. the "payment" received by user x_i^t in round t.

It is easy to verify that $\left|\tau^t\right| \leq 2\delta w_i^t$, so that each $w^t(x)$ changes by at most a factor of $(1 \pm 2\delta)$ in any round. Since $\delta < \frac{1}{4}$, this guarantees that $w^t(x) > 0$.

To analyze this algorithm we rely on the potential functions

$$U^T(x) = \frac{1}{\delta} \log\left(w^T(x)\right) + p^{<T}(x),$$

where $p^{<T}(x)$ is the total payoff received by user x over rounds $1, 2, \ldots T-1$ (and similarly for $p^{\leq T}(x)$). We use three lemmas to relate the reported payoffs p^t to the actual payoffs $p^{\leq T}(x)$.

Our first lemma shows that the modified payoffs are at least as large as the smaller of the two original payoffs. This lemma is why we need to use a symmetrized benchmark.

Lemma 8 $\mathbb{E}\left[p^t\right] \geq \min\left\{\mathbb{E}\left[p_0^t\right], \mathbb{E}\left[p_1^t\right]\right\}$

Proof We have

$$\begin{aligned}
\mathbb{E}\left[p^t\right] &= \mathbb{E}\left[\beta_0^t p_0^t + \beta_1^t p_1^t\right] \\
&= \beta_0^t \mathbb{E}\left[p_0^t\right] + \beta_1^t \mathbb{E}\left[p_1^t\right] \\
&\geq \min\left\{\mathbb{E}\left[p_0^t\right], \mathbb{E}\left[p_1^t\right]\right\}
\end{aligned}$$

as desired, where the last inequality follows from $\beta_0^t + \beta_1^t = 1$. ■

Our second lemma relates p^t to the change in $U^t(x)$. Essentially, it shows that both $U^t\left(x_0^t\right)$ and $U^t\left(x_1^t\right)$ increase by p^t in each round. This was the motivation for our definition of τ—it is a transfer which is exactly large enough to equate the payoffs of the two players.

Lemma 9 *If $s^t = 1$, then $U^{t+1}\left(x_i^t\right) \geq U^t\left(x_i^t\right) + p^t - \mathcal{O}(\delta)$.*

Proof We have

$$\begin{aligned}
U^{t+1}\left(x_i^t\right) &= \frac{1}{\delta}\log\left(w^{t+1}\left(x_i^t\right)\right) + p^{<t+1}\left(x_i^t\right) \\
&= U^t\left(x_i^t\right) + \frac{1}{\delta}\log\left(\frac{w^{t+1}\left(x_i^t\right)}{w_i^t}\right) + p_i^t \\
&= U^t\left(x_i^t\right) + \frac{1}{\delta}\log\left(1+\frac{\tau_i^t}{w_i^t}\right) + p_i^t \\
&> U^t\left(x_i^t\right) + \beta_{1-i}^t\left(p_{1-i}^t - p_i^t\right) + p_i^t - 4\delta \\
&> U^t\left(x_i^t\right) + \beta_{1-i}^t\left(p_{1-i}^t\right) + \left(1-\beta_{1-i}^t\right)p_i^t - 4\delta \\
&> U^t\left(x_i^t\right) + \beta_{1-i}^t\left(p_{1-i}^t\right) + \beta_i^t p_i^t - 4\delta \\
&= U^t\left(x_i^t\right) + p^t - 4\delta
\end{aligned}$$

where we have used the bound $\left|\frac{\tau_i^t}{w_i^t}\right| \leq 2\delta$ and the inequality $\log(1+z) > z - 4\delta^2$ for $|z| \leq 2\delta <<$ 12. ■

Our third lemma relates the quantities $U^T(x)$ to the payoffs $p^{\leq T}(x)$. Essentially, it shows that increases in users' balances cannot be responsible for sustainable increases in U. Over the very long term increases in U can only be due to users receiving high payoffs.

Lemma 10

$$p^{\leq T}(H) \geq \sum_{x\in H} U^T(x) - \frac{1}{\delta}\left|H\right|\log\left(\frac{N}{|H|}\right)$$

Proof $\sum_x w^t(x)$ is conserved by the algorithm, and $w^t(x) \geq 0$, so

$$\sum_{x\in H} w^T(x) \leq \sum_x w^T(x) = \sum_x w^0(x) = N.$$

So by Jensen's inequality, $\sum_{x\in H}\log\left(w^T(x)\right) \leq |H|\log\left(\frac{N}{|H|}\right)$. This implies:

$$\begin{aligned}
p^{\leq T}(H) &= \sum_{x\in H} U^T(x) - \frac{1}{\delta}\sum_{x\in H}\log\left(w^T(x)\right) \\
&\geq \sum_{x\in H} U^T(x) - \frac{1}{\delta}\left|H\right|\log\left(\frac{N}{|H|}\right),
\end{aligned}$$

as desired. ■

Combining these three lemmas yields our desired result: we know that p^t is at least as large as our benchmark, we know that U^T increases by p^t each round, and we know that in the long run increases in U^T are equivalent to increases in $p^{\leq T}$. The full proof deals with the approximation errors introduced by the curvature of the utility function, the deviation introduced by transient changes in balances, and the regret of the algorithm $\mathcal{A}$:

Theorem *[Restatement of Theorem 2] $\mathcal{A}_{reputation}$ obtains payoffs*

$$p^{\leq T}(H) \geq OPT^{\leq T}_{sym}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right).$$

Proof We will assume that $T > 16N$ is known in advance, and choose $\delta = \sqrt{\frac{N}{T}}$; this assumption can then be removed by a standard doubling trick exactly as in the proof of Theorem 1.

By applying Theorem 3 twice and Lemma 8, we have

$$\begin{aligned}\sum_{\substack{t,i:x_i^t \in H \\ s^t=1}} p^t &\geq \sum_{t:x_0^t,x_1^t \in H} \left(p^t\right) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right) \\ &= \sum_{t:x_0^t,x_1^t \in H} 2 * \min\left\{\mathbb{E}\left[p_0^t\right], \mathbb{E}\left[p_1^t\right]\right\} - \mathcal{O}\left(\sqrt[3]{T^2 N}\right).\end{aligned}$$

Then we can apply Lemma 9 and Lemma 10 and obtain in expectation

$$\begin{aligned}p^{\leq T}(H) &\geq \sum_{x \in H} U^T(x) - \frac{1}{\delta}|H| \log\left(\frac{N}{|H|}\right) \\ &\geq \sum_{x \in H} U^T(x) - \frac{N}{\delta} \\ &= \sum_{x \in H} U^T(x) - \mathcal{O}\left(\sqrt{NT}\right) \\ &\geq \sum_{x \in H} U^0(x) + \sum_{\substack{i,t:x_i^t \in H \\ s^t=1}} p^t - \delta T - \mathcal{O}\left(\sqrt{NT}\right) \\ &= \sum_{\substack{i,t:x_i^t \in H \\ s^t=1}} p^t - \mathcal{O}\left(\sqrt{NT}\right) \\ &\geq \sum_{t:x_0^t,x_1^t \in H} 2 * \min\left\{\mathbb{E}\left[p_0^t\right], \mathbb{E}\left[p_1^t\right]\right\} - \mathcal{O}\left(\sqrt[3]{T^2 N}\right) - \mathcal{O}\left(\sqrt{NT}\right) \\ &\geq \text{OPT}^{\leq T}(H) - \mathcal{O}\left(\sqrt[3]{T^2 N}\right)\end{aligned}$$

where the last inequality uses the assumption $T > N$. ■

Appendix E. Analysis of online local learning

Lemma 11 *(Christiano (2014)) For any $E^{<t}$ and any E^t we have*

$$\left|\mathcal{OLL}_\varepsilon\left(E^{<t}\right) - \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)\right|_\infty \leq \mathcal{O}\left(\varepsilon \left|E^t\right|_1\right)$$

Proof Note that the ℓ_1 and ℓ_∞ norms are dual. So this result follows directly from Lemma 8 in (Christiano (2014)) and Lemma 2.19 in (Shalev-Shwartz (2012)). ■

Theorem 12 *(Christiano (2014)) For any sequence E^t and any $X \succeq 0$ with entries in $[0,1]$, we have*

$$\sum_t \left\langle E^t, \mathcal{OLL}_\varepsilon\left(E^{<t}\right)\right\rangle \geq \sum_t \left\langle E^t, X\right\rangle - \mathcal{O}\left(\varepsilon \sum_{t<T} \left|E^t\right|_1^2\right) - \mathcal{O}\left(N\varepsilon^{-1}\right)$$

where $\left|E^t\right|_1$ is the sum of the absolute values of the entries of E^t.

Proof By Lemma 2.3 of (Shalev-Shwartz (2012)) and Lemma 5 of (Christiano (2014)), for any $X \succeq 0$ with entries in $[0,1]$ we have:

$$\sum_t \left\langle E^t, \mathcal{OLL}_\varepsilon\left(E^{<t}\right)\right\rangle \geq \sum_t \left\langle E^t, X\right\rangle - \mathcal{O}\left(N\varepsilon^{-1}\right) + \sum_t \left\langle E^t, \mathcal{OLL}_\varepsilon\left(E^{<t}\right) - \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)\right\rangle.$$

Using Lemma 11 we have

$$\begin{aligned}\sum_t \left\langle E^t, \mathcal{OLL}_\varepsilon\left(E^{<t}\right) - \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)\right\rangle &\geq -\sum_t \left|E^t\right|_1 \left|\mathcal{OLL}_\varepsilon\left(E^{<t}\right) - \mathcal{OLL}_\varepsilon\left(E^{\leq t}\right)\right|_\infty \\ &= \mathcal{O}\left(\sum_t \varepsilon \left|E^t\right|_1^2\right)\end{aligned}$$

which gives the desired result. ■

Appendix F. Open questions

We have presented a protocol for managing trust which achieves a qualitatively new—and surprisingly strong—formal guarantee. But the theoretical picture is far from complete, and the proposed protocol is far from practical. We hope that our model and algorithmic approach will inspire further work in both of these directions, and in this section we highlight some areas for improvement and elaboration.

Tighter regret bounds: Our regret bound of $\mathcal{O}\left(\sqrt[3]{T^2 N}\right)$ can probably be improved to $\mathcal{O}\left(\sqrt{TN}\right)$, which translates into a significant performance improvement.

When the fraction of dishonest users is ε, it may be possible to achieve a regret of $\mathcal{O}\left(\sqrt{TN\varepsilon\log(\varepsilon)}\right)$. For small values of ε, this would be major improvement over our current bound, which doesn't depend on the number of dishonest users. Similarly, when the fraction α of honest users is small, it may be possible to achieve regret of $\mathcal{O}\left(\sqrt{TN\alpha\log(\alpha)}\right)$. In some settings, this may be a very large improvement in regret per honest user.

Our regret bound depends on the total number of interactions. This might be improved to depend only on the number of interactions involving an honest user. The distinction is particularly important in settings where dishonest users can fabricate large numbers of interactions amongst themselves, or where the fraction of honest users is very small.

Distributed implementation: Although our protocol is is polynomial-time, it has no efficient distributed implementation. Ideally each user's computation and communication would be polylogarithmic in the size of the community.

The main challenge is distributing the online learning algorithm at the core of our protocol. The current algorithm requires semidefinite programming, which must be replaced by fast distributed computation, such as a random walk or sparse linear algebra. In addition to the algorithmic difficulties, the computation must be robust to manipulation by adversarial users; this might be achieved either by finding a computation which is inherently robust or by using cryptographic tools to ensure accurate computation.

Simultaneous interactions: Our protocol assumes that interactions occur one at a time: after one pair of users interacts, their reports immediately become available to the next pair of users. In most contexts this is not a realistic assumption.

When applied to a setting with k simultaneous interactions, our protocol achieves regret $\Omega(k)$. It is not hard to construct examples where the regret must necessarily be $\Omega(\log k)$. We suspect that this lower bound is tight, and that some protocol achieves regret $\mathcal{O}\left(\log k\sqrt{TN}\right)$. Under realistic assumptions, it may be possible to eliminate the dependence on k altogether.

Recommending vs. filtering: Our protocol answers questions of the form "should I interact with this merchant?" rather than questions of the form "which merchant should I interact with?" In many contexts the latter question is more useful—though deciding which users' input to trust is a key difficulty for both problems. (Awerbuch and Kleinberg (2005)) provides a similar algorithm that answers "which merchant should I interact with?" but exploits the simplifying assumption that there is a single best merchant for all transactions. It is natural to try to combine these two assumptions, and to design a protocol which recommends one resource from amongst k options.

Incentive compatibility: Our protocol has a robust theoretical guarantee, but that guarantee is only meaningful to the extent that users report meaningful payoffs and are willing to engage in productive interactions. Our protocol does not provide incentives for users to report honestly or to engage in interactions that benefit other users (except that each user must contribute enough that interacting with them is a net positive for the rest of the community).

A more sophisticated protocol might be able to provide such incentives. Because our environment is quite general we can easily embed problems like bilateral trade, and so any algorithms will need to deal with negative results such as Myerson-Satterthwaite. But even in light of these limits, there is likely to be room for a much deeper understanding of and accommodation for user incentives.

Asymmetric interactions: Our protocols all require that interactions be symmetric *on average*, or else make use of transfers to symmetrize them. This is a very strong condition that would be nice to weaken.

One generalization is to consider sets of interactions that are globally balanced, so that each user has as many opportunities to give as to receive overall even though the pairwise interactions are asymmetrical.

A further generalization is to sets of interactions that are balanced only when different users are assigned different weights. Any Pareto efficient allocation corresponds to maximizing some weighted combination of users' payoffs, and we could attempt to learn such weights in parallel with the rest of the reputation system problem. The key difficulty is doing this learning in a way that is robust to manipulation.

Sybilproofness: Our system bounds the damage done by the introduction of any dishonest user. In some settings it is easy for an attacker to create large numbers of dishonest users or *sybils*. If creating sybils is very cheap, then our bounds may not be tight enough to discourage such an attack on their own.

By extending the model it might be possible to ensure that the introduction of a dishonest user does *no* damage to the honest users. For example, if monetary transfers are available, each user could pay an upfront deposit to cover any possible damage they might inflict (in particular, a new user would need to pay a cost of at least 1 before *any* interaction with an established user, which would potentially be refunded if the interaction went well). Actually designing such a system appears to be quite subtle, but our tight bounds on the damage per dishonest user seems like a useful first step.

Composition and complex protocols: Worst-case performance guarantees allow our protocol to be used as a subroutine without compromising its correctness. Together with other building blocks, our result might be used to design more complex protocols in the same way that simple cryptographic primitives are used to build more complex cryptographic systems.