A Game-Theoretic Perspective on Trust in Recommendation

Sarah H. Cen [1] **Andrew Ilyas** [1] **Aleksander Madry** [1]

Abstract

Recommendation platforms—such as Amazon, Netflix, and Facebook—use various strategies in order to engage and retain users, from tracking their data to showing addictive content. These measures are meant to improve performance, but they can also erode their users' *trust*. In this work, we study the role of trust in recommendation. We show that, because recommendation platforms rely on users for data, trust is key to every platform's success. Our main contribution is a game-theoretic view of recommender systems and a corresponding formalization of trust. More precisely, if a user trusts their recommendation platform, then their optimal long-term strategy is to act greedily—and thus report their preferences truthfully—at all times. Our definition reflects the intuition that trust arises when the incentives of the user and platform are sufficiently aligned. To illustrate the implications of this definition, we explore two simple examples of trust. We show that distrust can hurt the platform and that trust can be beneficial for both the user and platform.

1. Introduction

YouTube, Google, and Facebook are all *recommendation platforms* whose success relies on their ability to recommend content that engages and retains users. To this end, these platforms employ *recommenders*, systems that present each user with a set of recommendations selected from a pool of content (Ricci et al., 2011; Resnick & Varian, 1997). Broadly, the study of recommenders is primarily concerned with finding the "best" content for users, and many systems take for granted that the data on which they are trained accurately reflect the users' content preferences.

In reality, users are not blind to how platforms operate. Instead, many learn how platforms work and adapt their behavior in order to get the experience that they want. A Spotify user might not "like" a song that they enjoy because they do not want to be recommended other songs by the same artist. Users who believe that a platform shares their data with advertisers might respond by browsing in Incognito mode (Klosowski, 2022). Users may adapt their behavior even when they love their recommendation algorithm (DeVito, 2019). For example, a user might avoid watching YouTube videos with friends who have different preferences in order to avoid tainting their own future recommendations.

These *strategic* behaviors violate a basic assumption of most recommenders; namely that their training data (e.g., what users choose to click on) genuinely depict the user's interests (Ekstrand & Willemsen, 2016; Stray et al., 2021). In the face of strategic behavior, platforms are left with a few options. They can use the strategic data and risk presenting suboptimal recommendations. Or platforms can (and indeed, do) try to anticipate and correct for their users' strategic behavior, often through more intense data collection, tracking, or personalization. However, there is no guarantee that the latter approach works or that users will not adapt yet again.

In this paper, we argue that a key force in this interaction is the *trust* between a user and their recommendation platform (Jacovi et al., 2021; Bose & Camerer, 2021). Specifically, when users don't trust their platform, they manipulate their behavior, corrupting the data that platforms collect. If platforms continue to erode this trust, both the platform and user suffer. This view suggests that instead of trying to anticipate how users adapt, then adjusting for this strategic behavior, recommenders should work *with* the user to build trust and make recommendation more *cooperative*.

While building trust is intuitively compelling, how do we make this goal more concrete? In this paper, we show how to formalize trust—casting recommendation as a two-player game between a user and their platform—and find that this game-theoretic lens opens paths for studying and building trustworthy recommendation platforms.

2. Model

We now introduce a game-theoretic model for studying recommender systems. Specifically, we cast recommendation as an alternating two-player game (Roth et al., 2010) be-

[1] Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Sarah H. Cen <shcen@mit.edu>.

ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments, Baltimore, Maryland, USA, 2022.

tween a user and their platform. We first define alternating two-player games in their full generality. We then specialize to the case of recommender systems, then define the user's and platform's strategies in this context. Finally, we build on this setup to provide an operational definition of *trust* between the user and their platform.

General setup. Two-player alternating games are specified by a pair of *action sets* (A_0, A_1) and a pair of *payoff functions* (U_0, U_1), where $U_i : A_0 \times A_1 \to [-1, 1]$ for $i \in \{0, 1\}$.[1] The game then proceeds in a series of steps, denoted by $t = 0, 1, 2, \ldots$ The action vector $(a^0_t, a^1_t) \in A_0 \times A_1$ captures both player's actions at time step t. On time steps where $t \pmod 2 = i$, player i plays a new action $a^i_t \in A_i$, and the other player plays their action from time step . At the end of each time step, player i collects the *payoff* $U_i(a^0_t, a^1_t)$.

Recommender systems. We now specialize this setup to recommender systems. Let players 0 and 1 be the platform p and user u respectively. On odd time steps t, the platform chooses a *recommender* $f_t \in \mathcal{F}$, where $\mathcal{F}$ is the set of all possible recommenders. In a simple case, $\mathcal{F}$ could be the power set of all available items on the platform, and f_t would be the set of items recommended at time step t. We can also capture more complex cases such as when f_t parameterizes a recommendation algorithm or captures details of the platform's user interface. On even time steps, the user chooses their behavior $b_t \in \mathcal{B}$, i.e., their actions on the platform. Once again, b_t might encode something as straightforward as whether the user clicks on each recommended item, but it can also capture more nuanced user behavior. For example, b_t might parameterize a function that describes how the user behaves.

We use $U_p : \mathcal{F} \times \mathcal{B} \to [-1, 1]$ to denote the payoff function of the platform. For instance, U_p might map from a set of recommendations $f \in \mathcal{F}$ and a set of clicks $b \in \mathcal{B}$ to a measure of user engagement (e.g., average clicks per recommended item). We similarly use $U_u : \mathcal{F} \times \mathcal{B} \to [-1, 1]$ to denote the payoff function of the user. The game is thus specified by $G = \{\mathcal{F}, \mathcal{B}, U_p, U_u\}$.

Strategies. The way that players act is captured by their *strategy*. Let $H_t = (f_\tau, b_\tau)_{\tau=0}^{t-1} \in \mathcal{H} = (\mathcal{F} \times \mathcal{B})^*$ denote the *history* of the game up to time t. A strategy for a player is a function mapping a history to the player's upcoming action. If the function is deterministic, the strategy is called *pure*. For notational clarity, we consider only pure strategies in this work, but our model and results extend straightforwardly to non-deterministic *(mixed)* strategies. Let $\mathcal{S}_p := \mathcal{H} \to \mathcal{F}$ and $\mathcal{S}_u := \mathcal{H} \to \mathcal{B}$ be the *pure strategy space* of the platform and user, respectively, i.e., the set of all deterministic functions mapping histories to actions for each player. We use $s = (s_p, s_u) \in \mathcal{S}_p \times \mathcal{S}_u$ to denote the (pure) strategies adopted by the platform and user. With slight abuse of notation, the *average payoff until time step* T for player $i \in \{p, u\}$ under a pair of strategies s is

$$P_{i,T}(s) = \frac{1}{T+1} \sum_{t=0}^{T} U_i(s(H^s_t)),$$

where H^s_t is the history generated by the repeated application of s and, for any $H_t \in \mathcal{H}$,

$$s(H_t) = \begin{cases} (s_p(H_t), b_{t-1}) & \text{if } t \pmod 2 = 1, \\ (f_{t-1}, s_u(H_t)) & \text{if } t \pmod 2 = 0. \end{cases}$$

Player i's *best response* strategy s^{BR}_i is one that maximizes their payoff in response to the other player's most recent action. Formally, the best response strategy of player i maps any $H_t \in \mathcal{H}$ to a strategy $s^{\text{BR}}_i(H_t)$ that satisfies

$$U_p(s^{\text{BR}}_p(H_t), b_{t-1}) \geq U_p(f, b_{t-1}) \qquad \forall\, f \in \mathcal{F},$$
$$U_u(f_{t-1}, s^{\text{BR}}_u(H_t)) \geq U_u(f_{t-1}, b) \qquad \forall\, b \in \mathcal{B},$$

where (f_{t-1}, b_{t-1}) is the last pair in H_t. Note that either player (or both players) can play a best response strategy, regardless of how the other player behaves.

Trust. We now turn towards the main objective of this section, which is to offer a concrete definition of *trust* in recommender systems. Our goal is for this definition to mirror intuitive, philosophical notions of trust, which we discuss at the end of this section. At a high level, a user trusts their platform if they behave truthfully towards it.

Definition 2.1. A user's strategy $s_u \in \mathcal{S}_u$ is H_t*-truthful* if and only if $s_u(H_t)$ is the user's best response to f_{t-1}, i.e.,

$$s_u(H_t) \in \arg\max_{b \in \mathcal{B}} U_u(f_{t-1}, b).$$

Definition 2.1 says that the user's behavior is truthful when their action is optimal for the user in the immediate term. In other words, the user is truthful if they act greedily in response to the current recommender f_{t-1}. The natural counterpart of a truthful (i.e., *short-term optimal*) strategy is a *long-term optimal* strategy, defined below.

Definition 2.2. Let $s_p \in \mathcal{S}_p$ denote the platform's strategy. Then, the user's *long-term optimal strategy* to s_p is

$$s^*_u \in \arg\max_{s_u \in \mathcal{S}_u} \lim_{T \to \infty} P_{u,T}(s_p, s_u).$$

Definition 2.2 says that s^*_u is the long-term optimal strategy to s_p when the average long-term payoff that follows from repeatedly playing the strategies (s_p, s^*_u) is optimal. The truthful and long-term optimal strategies do not necessarily

[1] Note that any bounded pair of utility functions can be rescaled to $[-1, 1]$ without loss of generality.

coincide, as the optimal action in the immediate term may lead to a suboptimal long-term trajectory if the platform responds poorly (e.g., if Twitter treats a user's outrage-driven retweet as strong interest in similar content). This observation leads to a natural definition of trust, as follows.

Definition 2.3 (Trust)**.** If a user *trusts* their platform's strategy $s_p \in \mathcal{S}_p$, then their optimal long-term strategy $s_u^* \in \mathcal{S}_u$ to s_p is H_t-truthful for every $t = 0, 1, 2, \ldots$

Definition 2.3 captures the intuition that, when a user trusts their recommendation platform, they can take actions that are good for them in the immediate term and trust that doing so will lead to good long-term outcomes for them, i.e., that those actions will not lead the platform to behave in ways that are harmful or suboptimal for the user. In particular, our formal definition mirrors intuitive notions of trust, such as Hardin's characterization of trust as a reflection of "encapsulated interest" (Hardin, 2002).

3. Unpacking Trust Through Examples

In this section, we build intuition for the game-theoretic perspective on trust provided in Section 2 by studying two (highly simplified) example settings.

In both examples, we consider the case where the user and platform both collect their payoffs every *other* round. In other words, both user and platform collect no payoff after each platform action; they only receive payoffs after the user acts. We anticipate that more complex examples (e.g., when the user has adapting preferences) can take advantage of our model's full generality.

3.1. A Privacy-Conscious User

In our first example, we consider a *privacy-conscious* user who is interested in protecting a particular sensitive attribute. For example, a user may wish to use Facebook without having Facebook learn the user's height. Unless the platform is designed with trust in mind, such a user is incentivized to manipulate their behavior to remove any correlation between their behavior and the sensitive attribute. We show that this leads to the recommender system learning an incorrect model of the user and providing poor recommendations in the long term. To circumvent this issue, the recommender system can *build trust* by playing a strategy that is suboptimal in the near-term but incentivizes the user to behave truthfully. In the long term, this approach leads to a better solution for both the user and platform.

3.1.1. Setup

We assume the universe $\mathcal{X}$ of items to recommend is given by the unit ball $\mathcal{B}(0, 1)$. We adopt a standard latent variable model such that the user's interest y_i in item i is a linear combination of the user's feature vector (i.e., preferences) θ and the item vector $x_i \in \mathbb{R}^d$ such that

$$y_i = \theta^\top x_i + \mathcal{N}(0, 1). \tag{1}$$

Actions. On its turn, the platform plays a *user model* $\hat{\theta}$, i.e., its estimate of the user's preferences θ, along with a set of k recommendations induced by this model. The user observes these k items and decides how much time to spend on each one. Their action is thus a vector $b \in \mathbb{R}^k$.

Payoffs. The platform's payoff is the total amount of time the user spends on recommended content, so $U_p(f, b) = \mathbf{1}^\top b$. Meanwhile, the user's payoff depends on both how much they can engage with recommendations that they like (where their enjoyment is quantified by (1)), and how well they are able to *hide* their i-th feature, i.e., the sensitive feature θ_i. The user's payoff is thus

$$U_u(f, b) = \sum_{j=1}^{k} \min(y_j, b_j) - \lambda \cdot \log(|\hat{\theta}_i - \theta_i|).$$

Strategies. For a given set of recommendations, the user's best response (which, per Definition 2.1, is the action given by their "truthful strategy") is to play $b = y$.[2] When the user plays truthfully, the platform's problem simplifies to that of learning with bandit feedback, a commonly studied recommmendation setting. A reasonable approach here is to play a learned user model

$$\hat{\theta}^{(t)} = \arg\min_{\theta'} \sum_{\tau=0}^{t} \sum_{i=1}^{k} (x_i^\top \theta' - b_{t,i})^2, \tag{2}$$

and to generate the induced items via the BALLEXPLORE algorithm (Deshpande & Montanari, 2012). Deshpande & Montanari (2012) show—again, *if the user plays truthfully*—that this strategy is long-term optimal for the platform.

3.1.2. The Role of Trust

So far, we have shown that if the user plays according to the best response strategy (that is, *truthfully*) then the natural strategy (2) is optimal for the platform. If the user does not care about hiding their private feature (i.e., if $\lambda = 0$), then the combination of (2) from the platform and truthful play $b = y$ from the user is also *long-term user optimal*.

Poor outcomes under truthfulness. When the user cares about hiding a feature, however, it turns out that truthful strategy is *suboptimal* for their long-term reward. The user is instead incentivized to behave strategically, unless the recommender system tries to build trust.

Specifically, when $\lambda > 0$, truthful play by the user will lead to the platform learning $\hat{\theta} = \theta$ (and in particular $\hat{\theta}_i = \theta_i$), and so the user's long-term reward will diverge to $-\infty$.

[2] We note that y can have negative elements. For ease of notation, we allow "negative time".

Strategic behavior. Thus, when the platform uses (2) the user is incentivized to behave *strategically* in lieu of playing its best-response strategy. For example, the user might skip over content where the feature is too prominent (i.e., report $b_j = 0$ for any item x_j where $x_{ji} \geq \kappa$ for some $\kappa > 0$). This strategic behavior corrupts the platform's estimate of $\hat{\theta}_p$. Although it improves privacy, it does so at the cost of lowering the quality of *all* recommendations.

Building trust. To repair trust, the platform can alter s_p to only show the user items where $x_{ip} = 0$. From a greedy perspective, this is suboptimal for the platform—it will be unable to learn $\hat{\theta}_p$ from the user's behavior. On the other hand, the platform eliminates the regularization term in the user's objective (since $x_{ip} - \overline{x}_{\cdot p} = 0$), which re-incentivizes the user to play truthfully.

3.2. Multimodal Preferences

In this section, we study a second setting in which a user has different preferences based on their mood. In this setting, the user learns that acting truthfully (i.e., according to their mood) results in poor recommendations when the platform is not equipped to handle the user's multimodal preferences. Instead, the user is incentivized to behave strategically, revealing only one of their moods to the platform. We conclude by showing that allowing users to explicitly indicate their mood builds trust and results in better outcomes for both the user as well as the platform.

3.2.1. Setup

Actions. Consider a platform that recommends movies. For simplicity, suppose that, at every time step $t \in [T]$, the platform recommends k movies $f_t \in [-1,1]^{k \times d} = \mathcal{F}$ to the user such that $f_{t,i} \in [-1,1]^d$ is the i-th movie's feature vector. In this example, let $k = 3$ and $d = 1$. Let $f_{t,i} = -1$ indicate a silly comedy, $f_{t,i} = 1$ indicate a serious drama, $f_{t,i} = 0$ indicate a movie that is equal parts comedy and drama, and so on. Given k recommendations, let $b_t \in \{0,1\}^k = \mathcal{B}$, where $b_{t,i} = 1$ indicates that the user watches the i-th movie at time t, and $b_{t,i} = 0$, otherwise.

Payoffs. Let $\theta_t \in [-1,1]$ denote the user's preference at time t, where $\theta_t = -1, +1$ correspond to comedies and dramas, respectively. Recall that the platform and user alternate turns. Let the user's payoff at time t be

$$U_u^t(f_t, b_t) = \frac{1}{k} \sum_{i=1}^{k} b_{t,i} \left(\mathbb{1}\{f_{t,i} = \theta_t\} - \frac{1}{2} \right).$$

The platform seeks to maximize engagement, so its payoff is: $U_p(f_t, b_t) = \sum_{i=1}^{k} b_{t,i}$.

Strategies. Suppose that the user is always in one of two moods $\theta_t \in \{-1, 1\}$, where $\theta_t \sim \text{Rad}(p)$ for $p \in [0,1]$. For a given set of recommendations f_t, the user's best response (i.e., *truthful*) strategy is to click only the content that matches their mode, i.e., to play $b = \mathbb{1}\{f_{t-1,i} = \theta_t\}_{i=1}^k$.

As for the platform, we consider the set of strategies wherein the platform estimates the user's preferences by observing their behavior, then recommends based on its estimate.

3.2.2. The role of trust

We now show that, unless the platform works with the user, they both receive worse outcomes than when they cooperate.

Poor outcomes under truthfulness. Suppose that the user is truthful and that the platform models the user as having only a single preference $\theta \in [-1,1]$. Let $\hat{\theta}_t \in [-1,1]$ denote the platform's estimate of θ at time t. In this case, truthfulness leads to undesirable user outcomes (incentivizing strategic behavior) in two ways, given next.

(1) *Feedback loops*. Suppose $f_0 = (-1, 0, 1)$ and for $t > 0$, the platform computes the maximum likelihood estimate $\hat{\theta}_t$ for θ, then recommends $f_{t,i} \sim \text{clip}(\mathcal{N}(\hat{\theta}_t, 1), -1, 1)$. In this case, the results of Hashimoto et al. (2018) imply that $\hat{\theta}$ will diverge to either -1 or 1 (depending on p), meaning the platform will only cater to one mood. As a result, the user's optimal strategy is to use the recommender only when they are in the higher-probability mood.

(2) *Incorrect model of preferences*. Suppose the platform is able to avoid feedback loops by adopting a robust estimation strategy and learning $\hat{\theta}_t = p$. Still, because the platform does not account for the user's moods, the platform learns bimodal preferences as unimodal. Instead of learning that the user likes comedies *or* dramas, the platform thinks $\hat{\theta}_t = p$, i.e., that the user likes movies with both comedy and drama. Since the user only likes pure comedies and pure dramas, the user's payoff is low when they are truthful.

Strategic behavior. In both cases above, the user's long-term optimal stragegy is to be strategic, i.e., untruthful, which by Definition 2.3, implies the user does not trust their platform. This result matches our intuition: the user manipulates their behavior because they do not trust that the platform understands truthful actions. Although the platform can "correct" for this by learning bimodal preferences $\theta = (\theta_1, \theta_2)$, the user's mood at any given time is still unknown, so the platform must either guess the user's mood or divide its recommendations across the two possible moods.

Building trust. The platform can build trust by allowing the user to explicitly express their mood and learning separate preferences for each mood. In this case, the user's long-term optimal strategy is to be truthful. Moreover, working with the user is also beneficial for the platform because it does not have to spend any recommendations on whatever mood the user is not in, nor does the platform need to devote effort into guessing the the user's mood, allowing the platform to fully engage the user based on their current preference.

References

Bose, D. and Camerer, C. Trust and behavioral economics. *The Neurobiology of Trust*, pp. 36, 2021.

Deshpande, Y. and Montanari, A. Linear bandits in high dimension and recommendation systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1750–1754. IEEE, 2012.

DeVito, M. A. User adaptation to constant change in algorithmically-driven social platforms. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, pp. 1–6, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359719. doi: 10.1145/3290607.3299082. URL https://doi.org/10.1145/3290607.3299082.

Ekstrand, M. D. and Willemsen, M. C. Behaviorism is not enough: better recommendations through listening to users. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 221–224, 2016.

Hardin, R. *Trust and trustworthiness*. Russell Sage Foundation, 2002.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pp. 1929–1938. PMLR, 2018.

Jacovi, A., Marasović, A., Miller, T., and Goldberg, Y. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 624–635, 2021.

Klosowski, T. How a burner browser hides my most embarrassing internet searches, 2022. URL https://www.nytimes.com/wirecutter/blog/burner-browser-to-hide-internet-searches/.

Resnick, P. and Varian, H. R. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Ricci, F., Rokach, L., and Shapira, B. Introduction to recommender systems handbook. In *Recommender systems handbook*, pp. 1–35. Springer, 2011.

Roth, A., Balcan, M. F., Kalai, A., and Mansour, Y. On the equilibria of alternating move games. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 805–816. SIAM, 2010.

Stray, J., Vendrov, I., Nixon, J., Adler, S., and Hadfield-Menell, D. What are you optimizing for? aligning recommender systems with human values. *arXiv preprint arXiv:2107.10939*, 2021.