Designing Optimal Binary Rating Systems

Nikhil Garg
Stanford University
nkgarg@stanford.edu

Ramesh Johari
Stanford University
rjohari@stanford.edu

Abstract

Modern online platforms rely on effective rating systems to learn about items. We consider the optimal design of rating systems that collect binary feedback after transactions. We make three contributions. First, we formalize the performance of a rating system as the speed with which it recovers the true underlying ranking on items (in a large deviations sense), accounting for both items' underlying match rates and the platform's preferences. Second, we provide an efficient algorithm to compute the binary feedback system that yields the highest such performance. Finally, we show how this theoretical perspective can be used to empirically design an implementable, approximately optimal rating system, and validate our approach using real-world experimental data collected on Amazon Mechanical Turk.

1 Introduction

Rating and ranking systems are everywhere, from online marketplaces (e.g., 5 star systems where buyers and sellers rate each other) to video platforms (e.g., thumbs up/down systems on YouTube and Netflix). However, they are uninformative in practice (Nosko and Tadelis, 2015). One recurring pattern is that ratings *binarize* – most raters only use the extreme choices on the rating scale, and the vast majority of ratings receive the best possible rating. For example, 75% of reviews on Airbnb receive a perfect rating of 5 stars (Fradkin et al., 2017). Furthermore, several platforms have adopted a binary rating system, in which a user rates her experience as either positive or negative. Given the prevalence of binary feedback (either de facto or by design), in this work we investigate the optimal *design* of such binary rating systems so that the platform can learn as fast as possible about the items being rated.

The rating pipeline often works as follows: A buyer enters a platform and *matches* with an item (e.g. selects a video on Youtube, is paired with a driver on Uber, or selects a home on AirBnB). She has an experience (e.g. a view, ride, or stay). Then, the platform asks her to *rate* her experience, i.e. it asks her a question. In a binary system, she indicates whether her experience was positive or negative. She then leaves. The platform uses the ratings it has received to score the quality of items, potentially showing such scores to future buyers.

By *designing* such a system, we mean: the platform can influence how the buyer rates – how likely she is to give a positive rating, conditional on the quality of her experience. It can do so by asking her different questions, e.g. "Was this experience above average" or "Was this experience the worst you've ever had?". Different questions shift the probabilities at which items of various qualities receive positive ratings.

Our first question is: *what is the structure of optimal binary feedback?* A rating system in which every buyer gives positive ratings after each match, independent of item quality, will fail to learn anything about the items. Clearly, better items should be more likely to receive positive ratings than worse ones. But how much more likely?

Informally, suppose we have a set of items that match with buyers over time (at potentially differing rates), and we wish to rank the items by their true quality $\theta_i \in [0, 1]$. The platform cannot observe θ_i, however. Rather, in our model, after each match, an item with quality θ_i receives a positive rating with probability $\beta(\theta_i)$, and negative otherwise. In

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

other words, the platform observes, for each item i, a sequence of ratings that are each Bernoulli($\beta(\theta_i)$). Such ratings are the only knowledge the platform has about items. The platform ranks the items according to the percentage of its ratings (samples) that were positive. The function $\beta : [0,1] \mapsto [0,1]$ affects how quickly the platform learns the true ranking, and it prefers to maximize the learning rate. We show how to calculate an optimal β.

As an example, consider three items with qualities $\theta_a > \theta_b > \theta_c$, and β such that $\beta(\theta_a) = 0.5$ and $\beta(\theta_c) = 0.1$, i.e. item a gets positive ratings after 50% of its matches, and item c after 10% of its matches. It is unclear what $\beta(\theta_b)$ should be. Trivially, $.1 < \beta(\theta_b) < .5$. Otherwise, even with infinitely many ratings the items will be mis-ranked.

But can we be more precise? If $\beta(\theta_b) = .49$, it will take many ratings of both items a and b to learn that $\theta_a > \theta_b$, but only a few from c to learn that $\theta_c < \theta_b$. That may be good if the platform wants to identify the worst item, but not if it wants to identify the best. It may also be fine if items a and b match much more often with buyers than item c. Clearly, the optimal value for $\beta(\theta_b)$ is objective and context dependent. Of course, the problem becomes more challenging with more items i for which $\beta(\theta_i)$ must be chosen. Lastly, in this example, one might intuitively think $\beta(\theta_b) = 0.3$ is optimal by symmetry when the items matter equally and matching rates are identical. This guess is incorrect. The optimal is $\beta(\theta_b) \approx 0.28$, due to the nature of binomial variance.

In this work, we first formalize the above problem and show how to find an optimal $\beta(\theta)$, jointly for a set of items $[0,1]$. β changes with the platform's objective and underlying item matching rates. Jumping ahead, Figure 1 shows optimal β in various settings under our model. For a platform that wants to find the worst sellers, for example, the top half of items should each get positive ratings at least $80+\%$ of the time; it is more important for the bottom half of items to be separated from one another, i.e. get positive ratings at differing percentages.

Once we have calculated the optimal rating function β (given context on the platform goals and matching rates), what should we do with it?

Our second question is: *How does a platform build a rating system such that buyers behave near-optimally, i.e. according to a calculated β?* The platform cannot directly control buyer rating behavior. Rather, it has to ask *questions* such that, for each item quality θ, a fraction $\beta(\theta)$ of raters will give the item a positive rating. For example, by asking, "Is this the best experience you've had," the platform would induce behavior such that $\beta(\theta)$ is small for most θ. Most platforms today ask vague questions (e.g. thumbs up/down), and items mostly get positive ratings. We show this is highly suboptimal for ranking items quickly.

Our main contributions and paper outline are:

Rating system design as information maximization. In Sections 3.1-3.2, we formulate the design of rating systems as an information maximization problem. In particular, a good rating system recovers the true ranking over items, and converge quickly in the number of ratings.

Computing an optimal rating feedback function β. In Section 3.3, we develop an efficient algorithm that calculates the optimal rating function β, which depends on matching rates and the platform objective. The optimal β provides quantitative insights and principled comparisons between designs.

Real-world system design. In Section 4, we show how a platform can use a simple experiment and existing data to empirically design a near-optimal rating system, and to audit the current system. In Section 5, we demonstrate the value of this approach through an experiment on Mechanical Turk.

2 Related work

Many empirical and model-based works document and tackle challenges in existing rating systems (Bolton et al., 2013; Cabral and Hortasu, 2010; Cook, 2015; Filippas et al., 2017; Fradkin et al., 2017; Gaikwad et al., 2016; Hu et al., 2009; Immorlica et al., 2010; Nosko and Tadelis, 2015; Rajaraman, 2009; Tadelis, 2016; Zervas et al., 2015). To our knowledge, we are the first to formalize a rating system design problem and then show how one can use empirical data to optimize such systems. In a related paper (Garg and Johari, 2018), we test behavioral insights using an experiment on a large online labor platform and develop a related design problem for a multiple-choice system, which proves far less tractable.

Other works also optimize platform learning rates (Acemoglu et al., 2017; Besbes and Scarsini, 2018; Che and Horner, 2015; Ifrach et al., 2017; Johari et al., 2017; Papanastasiou et al., 2017). When prescriptive, they modify *which matches occur*, while we view the matching process as given and modify *the rating system*. The solutions are complementary.

Many bandits works also seek to rank items from a sequence of observations (Katariya et al., 2016; Maes et al., 2011; Radlinski et al., 2008; Yue and Joachims, 2009). Our problem is the *inverse* of the bandit setting: given an arm-pulling policy, we design each arm's feedback.[1] Our specific theoretical framework is similar to that of Glynn and Juneja (2004), who optimize a large deviations rate to derive an arm-pulling policy for best arm identification.

The "twenty questions" interpretation of Shannon entropy (Cover and Thomas, 2012; Dagan et al., 2017) seeks questions that can identify an item from its distribution. Dagan et al. (2017) show how to almost match the performance of Huffman codes with only comparison and equality questions. Our work differs in two key respects: first, we seek to rank a set of items as opposed to identifying a single item; second, we consider non-adaptive policies (i.e. the platform cannot change its rating form in response to what it knows about an item already).

3 Model and optimization

We now formalize our model and show how to optimize the rating function to maximize the learning rate. We focus on finding an optimal $\beta : [0,1] \mapsto [0,1]$, a map from item quality θ to the probability it should receive a positive rating. This section requires no data: we characterize the *optimal* system.

3.1 Model and problem specification

Our model is constructed to emphasize the rating system's learning rate. Time is discrete ($k = 0, 1, 2, \ldots$). Informally: there is a set of items. Each time step, buyers match with the items and leave a rating according to $\beta(\theta)$. The platform records the ratings and ranks the items. Formally:

Items. The system consists of a set $[0,1]$ of items, where each item is associated with a unique (but unknown) quality $\theta \in [0,1]$; i.e., the system consists of a *continuum* of a unit mass of items whose unknown qualities are uniform[2] in $[0,1]$. Below, we *discretize* the continuous quality space $[0,1]$ into M types, to calculate a stepwise increasing β. We will make clear why we introduce a continuum but then discretize.

Matching with buyers. Items accumulate ratings over time by matching with buyers. We assume the existence of a nondecreasing *match function* $g(\theta)$, where item θ receives $n_k(\theta) = \lfloor kg(\theta) \rfloor$ matches, and thus ratings, up to time k. In other words, item θ is matched approximately every $\frac{1}{g(\theta)}$ time steps. $g(\theta) \leq 1$ and bounded away from 0, i.e. $\exists c > 0$: $g(\theta) > c$. This accumulation captures the feature that better items may be more likely to match.

Ratings. The key quantity for our subsequent analysis is the *probability of a positive rating* for each θ, $\beta(\theta) \triangleq \Pr(\text{positive rating}|\theta)$. Let $y_\ell(\theta) \sim \text{Bernoulli}(\beta(\theta))$ be the rating an item of quality θ receives at the ℓth time it matches.

Aggregating ratings and ranking sellers. These ratings are aggregated into a *reputation score*, $x_k(\theta)$, at each time k. The score is the fraction of positive ratings received up to time k: $x_k(\theta) \triangleq \frac{1}{n_k(\theta)} \sum_{\ell=0}^{n_k(\theta)} y_\ell(\theta)$ with $x_0(\theta) \triangleq 0$ for all θ. Thus, $x_k(\theta) \sim \frac{1}{n_k(\theta)} \text{Binomial}(\beta(\theta), n_k(\theta))$.

System state. The state of the system is given by a joint distribution $\mu_k(\Theta, X)$, which gives the *mass* of items of quality $\theta \in \Theta \subset [0,1]$ with aggregate score $x_k(\theta) \in X \subset [0,1]$ at time k. Because our model is a continuum, the evolution of the system state μ_k follows a deterministic dynamical system.

We have described these dynamics at the level of individual items; however, such statements should be interpreted as describing the evolution of the joint distribution μ_k. The state update for μ_k is determined by the *mass* of items that match and the *distributions* of their ratings. A formal description of the state evolution is in Appendix Section B.1.

Platform objective. The platform wishes to rank the items accurately. Given β and $\theta_1 > \theta_2$, define:

$$\begin{aligned} P_k(\theta_1, \theta_2|\beta) =& \mu_k\big(x_k(\theta_1) > x_k(\theta_2)|\theta_1, \theta_2\big) \\ &- \mu_k\big(x_k(\theta_1) < x_k(\theta_2)|\theta_1, \theta_2\big) \end{aligned} \quad (1)$$

This expression captures observed score ranking's accuracy. When $\theta_1 > \theta_2$ but $x_k(\theta_1) < x_k(\theta_2)$, the ranking mistakenly orders θ_1 below θ_2. A good system has large $P_k(\theta_1, \theta_2|\beta)$. Integrating across items creates the following objective for each time k:

$$W_k = \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) P_k(\theta_1, \theta_2|\beta) d\theta_1 d\theta_2 \quad (2)$$

Weight function $w(\theta_1, \theta_2) > 0$ indicates how much the platform cares about not mistaking a quality θ_1 item with a quality θ_2 item. We consider scaled w such that $\int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) d\theta_1 d\theta_2 = 1$.

Our first question then becomes: *What β yields the highest value of W_k?* As discussed above, the plat-

[1] Note that a *rating* is not the same as a *reward*; buyers often give positive ratings after bad experiences.

[2] Any distribution can be handled by considering θ to be the item's *quantile* rather than its absolute quality.

form influences β through the design of its rating system. The optimal choice of β sets the benchmark.

Discussion. *Objective function.* The specification (2) of the objective is quite rich. It contains scaled versions of Kendall's τ (with $w(\theta_1, \theta_2) = 1$ for all θ_1, θ_2) and Spearman's ρ (with $w(\theta_1, \theta_2) = \theta_1 - \theta_2$) rank correlations. w allows the platform to encode, for example, that it cares more about correctly ranking just the very best, very worst, or items at both extremes.[3] Tarsitano (2009) and da Costa and Roque (2006) discuss other well-studied examples.

Relationship between model components. Qualitatively, β affects W_k as follows, as previewed in the introduction: when $\beta(\theta_1) \approx \beta(\theta_2)$, then $x_k(\theta_1) \approx x_k(\theta_2)$, and so $P_k(\theta_1, \theta_2|\beta)$ is small (errors are common). A good design thus would have large $\beta(\theta_1) - \beta(\theta_2)$ for $\theta_1 > \theta_2$ where $w(\theta_1, \theta_2)$ is large. Matching function g also affects P_k and thus W_k: when $g(\theta)$ is large, more ratings are sampled from item of quality θ, i.e. $n_k(\theta)$ is higher, and so $x_k(\theta)$ is more closely concentrated around its mean $\beta(\theta)$. Thus, $P_k(\theta, \theta'|\beta)$ increases (for all θ') with $g(\theta)$. A good design of β thus considers both w and g.

Matching. As noted above, we assume items receive a non-decreasing number of ratings based on their true quality, through matching function $g(\theta)$. This is a reasonable approximation for our analysis, where we focus on the asymptotic rate of convergence of the ranking based on to the true ranking, as the number of ratings increases. In practice, items will be more likely to match when they have a higher *observed* aggregate score. Similarly, our model makes the stylized choice that all items have the same age. In reality, items have different ages in platforms.

Non-response. In practice, many buyers choose not to rate items, which our model does not capture. One possible approach is to treat non-response as a bad experience, which yields more information in the work of Nosko and Tadelis (2015). Solutions to non-response is an important area of work.

3.2 Large deviations & discretization

Recall the question: *What β yields the highest value of W_k?*. We now refine objective W_k and constrain β to form a non-degenerate, feasible optimization task.

Large deviation rate function. W_k is not one objective: it has a different value per time k, and no single β simultaneously optimizes W_k for all k.[4] Considering *asymptotic* performance is also insufficient: when β is strictly increasing in θ, $\lim_{k\to\infty} P_k(\theta_1, \theta_2|\beta) = 1 \ \forall \theta_1, \theta_2$ by the law of large numbers. Thus, $W \triangleq \lim_{k\to\infty} W_k = 1$, and *any* such β is asymptotically optimal.

For this reason, we consider maximization of the *rate* at which W_k converges, i.e., how *fast* the estimated item ranking converges to the true item ranking. We use a large deviations approach (Dembo and Zeitouni, 2010) to quantify this convergence rate. Formally, given sequence $Y_k \leq \lim_{k\to\infty} Y_k = Y$, the *large deviations rate of convergence* is $-\lim_{k\to\infty} \frac{1}{k} \log(Y - Y_k) = c$. If c exists then Y_k approaches Y exponentially fast: $Y - Y_k = e^{-kc+o(k)}$.

Then, we wish to *choose β to maximize W_k's large deviations rate*, $r = -\lim_{k\to\infty} \frac{1}{k} \log(W - W_k)$.

Discretizing β. Unfortunately, even this problem is degenerate if we consider continuous β: for any β that is not piecewise constant, the large deviations rate of convergence is zero, i.e., convergence of W_k to its limit is only polynomially fast, and characterizing the dependence of this convergence rate on β is intractable. Thus, the rate of convergence for W_k is not a satisfactory objective with continuous β.

We make progress by *discretizing* β; in particular, we restrict attention to optimization over stepwise increasing β functions.[5] Among stepwise increasing β, the large deviations rate of W_k to its limiting value W can be shown to be nondegenerate, i.e. $\exists c > 0$ s.t. $W - W_k = e^{-kc+o(k)}$. (See Lemma C.4 in the Appendix for further discussion.)

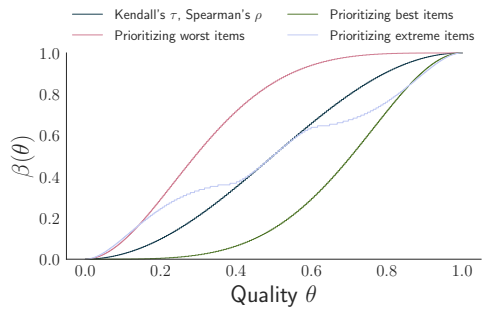
Notationally, we will calculate an optimal stepwise increasing β with M levels, i.e. there are M intervals $S_i \subset [0, 1]$ and levels t_i such that when $\theta_1, \theta_2 \in S_i$, then $\beta(\theta_1) = \beta(\theta_2) \triangleq t_i$. The challenge is calculating an optimal $\boldsymbol{S^*} = \{S_i\}$ and $\boldsymbol{t^*} = \{t_i\}$.

The physical interpretation is that we group the items into M subsets (types) $S_i \subset [0, 1]$. When items θ_1, θ_2 are in the same subset, then their asymptotic reputation scores are the same, $\lim_k x_k(\theta_1) = \lim_k x_k(\theta_2) \triangleq t_i$. These items cannot be distinguished from one another even asymptotically.

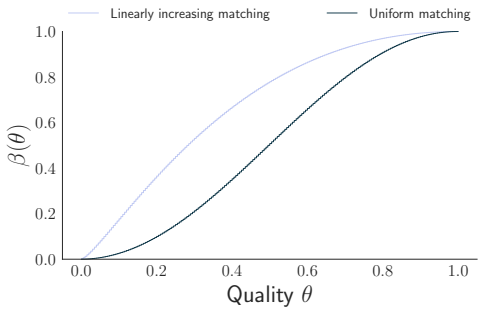
[3] We use $\theta_1\theta_2(\theta_1 - \theta_2)$, $(1 - \theta_1)(1 - \theta_2)(\theta_1 - \theta_2)$, and $(\frac{1}{2} - \theta_1)^2(\frac{1}{2} - \theta_2)^2(\theta_1 - \theta_2)$ as examples.

[4] For example, consider β such that the worst half of items never receive a positive rating and the rest always do. It would perform comparatively well for a small number of ratings k, as it quickly distinguishes the best from the worst items. However it would never distinguish items within the same half. Some β' may make more mistakes initially but perform better at larger k.

[5] Note that, even for purely computational reasons, calculating β requires discretization.



(a) Fix $g = 1$, with various objective function weights w



(b) Fix $w = 1$. Vary matching, $g = 1$, and $g = \frac{1+10\theta}{11}$

Figure 1: Optimal β (with $M = 200$) with various objective weight functions w and matching rates g.

Though discretization allows us to define a large deviations rate for W_k, it comes at a cost: W, the limiting value of W_k, is no longer one. Different discretization choices $\boldsymbol{S}$ result in different W.

Our optimization problem: *Within the class of stepwise increasing functions with M levels, find the β that is optimal, i.e. is*

(1) *Asymptotically optimal.* It yields the highest limiting value of W_k. AND

(2) *Rate optimal.* It yields the fastest large deviations rate r among *asymptotically optimal* β.

A remarkable result of our paper is a $\mathcal{O}\left(M \log^2 \frac{M}{\epsilon}\right)$ procedure to find an optimal β with M levels.

3.3 Solving the optimization problem

The theorem below shows that the problem decomposes into two stages: first, find optimal discretization intervals $\boldsymbol{S^*}$; then, find optimal $\boldsymbol{t^*}$ given $\boldsymbol{S^*}$.

Theorem 3.1. *The β defined by the following choices of $\boldsymbol{S}^*$ and $\boldsymbol{t}^*$ is optimal:*

$\boldsymbol{S}^* = \arg\max \sum_{0 \le i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2)$

$\boldsymbol{t}^* = \arg\max r(\boldsymbol{t})$, *where*[6] $g_i \triangleq \inf_{\theta \in S_i} g(\theta)$ *and*

$$r(\boldsymbol{t}) \triangleq -\lim_{k\to\infty} \frac{1}{k} \log(W - W_k) \tag{3}$$
$$= \min_{0 \le i \le M-2} \inf_{a \in \mathbb{R}} \{g_{i+1} KL(a||t_{i+1}) + g_i KL(a||t_i)\}$$

The proof is in the Appendix. The main hurdle is showing that the continuum of rates for $P_k(\theta_1, \theta_2)$ for each pair θ_1, θ_2 translates into a rate for W_k.

[6] $\mathrm{KL}(a||b) = a \log \frac{b}{a} + (1-a) \log \frac{1-b}{1-a}$ is the Kullback-Leibler (KL) divergence between Bernoulli random variables with success probabilities a and b respectively.

This decomposition separates our two questions: $\boldsymbol{S}^*$ maximizes the limiting value of W_k *given any* $\boldsymbol{t}$, and depends only on w; Then, $\boldsymbol{t}^*$ maximizes the rate at which the limiting value is reached, given g_i.

For Kendall's τ and Spearman's ρ, the optimal intervals are simply equispaced in $[0, 1]$, i.e. $S_i^* = [\frac{i}{M}, \frac{i+1}{M})$, because the entire item quality distribution is equally important. For other objective weight functions w, the difficulty of finding the optimal subsets $\boldsymbol{S}^*$ depends on the properties of w. Since $\boldsymbol{S^*}$ is trivial for Kendall's τ and Spearman's ρ – and w is just an analytic tool that formalizes a platform's goals – we focus on finding the optimal levels $\boldsymbol{t}^*$.

Discussion. One may naturally wonder why we introduced a continuum of quality $[0, 1]$ and then discretized into M subsets, instead of starting with M types. As established in Theorem 3.1, *how* we discretize (i.e. solving for $\mathbf{S}^*$) allows for optimization of different objective weight functions w; it determines *which* items are most valuable to distinguish.

Suppose we started with a set of M items. Then the only remaining challenge is to equalize the rates at which each item is separated from others: the large deviations rate is unaffected by the weight function w (it does not appear in the simplification of $r(\boldsymbol{t})$). In other words, *given* a discrete set of M items (equiv, given $\boldsymbol{S^*}$), calculating the optimal $\boldsymbol{t}$ is equivalent to solving a *maximin* problem for the rates at which each type is distinguished from each of the others. Thus, the algorithm below also solves the *inverse* bandits problem in which we wish to rank the M arms, and we can choose the structure of the (binary) observations at each arm.

We further note that the choice of M is not consequential; in the Appendix Section B.4 we show that in an appropriate sense, a sequence of optimal β_M for each M converges as M gets large.

Algorithm to find the optimal levels We now describe how to find $\boldsymbol{t}^*$, the maximizer of $r(\boldsymbol{t})$.

The following lemma describes a system of equations to find the $\boldsymbol{t}^*$ that maximizes r. It states that $\boldsymbol{t}^*$ equalizes the rates at which each interval i is separated from its neighbors. The proof involves manipulation of r and convexity, and is in the appendix.

Lemma 3.1. *The unique solution $\boldsymbol{t}^*$ to the following system of equations maximizes $r(\boldsymbol{t})$:*

$$r(t_0,t_1) = r(t_1,t_2) = \cdots = r(t_{M-2},t_{M-1}) \tag{4}$$
$$t_0 = 0, t_{M-1} = 1$$
$$r(t_{i-1},t_i) \triangleq -\log\left[\left((1-t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}}(1-t_i)^{\frac{g_i}{g_{i-1}+g_i}} + t_{i-1}^{\frac{g_{i-1}}{g_{i-1}+g_i}} t_i^{\frac{g_i}{g_{i-1}+g_i}}\right)^{g_{i-1}+g_i}\right]$$

We do not know of any algorithm that efficiently and provably solves such convex equality systems in general. However, we leverage some structure in our setting to develop an algorithm, *NestedBisection*, with run-time and optimality guarantees. The efficiency of our algorithm results from the property that, given a rate, t_i is uniquely determined by the value of either of the adjacent levels t_{i-1}, t_{i+1}, reducing an exponentially large search space to an almost linear one. Physically, i.e., we only need to separate each type of item from its neighboring types.

Below we include pseudo-code. Akin to branch and bound, the algorithm proceeds via bisection on the optimal value of t_{M-2}. For each candidate value of t_{M-2}, the other values can be found using a sequence of bisection subroutines. These values approximately obey all the equalities in the system (4) except the first. The direction of the first equality's violation reveals how to change the interval for the next outer bisection iteration.

Theorem 3.2. *NestedBisection finds an ϵ-optimal $\boldsymbol{t}$ in $\mathcal{O}\left(M \log^2 \frac{M}{\epsilon}\right)$ operations, where ϵ-optimal means that $r(\boldsymbol{t})$ is within additive constant ϵ of optimal.*

The proof is in the appendix. The main difficulty is finding a Lipschitz constant $\epsilon(\delta)$ for how much the rate changes with a shift δ in a level t_i. This requires lower bounding t_1 as a function of M. In practice, the algorithm runs instantaneously on a modern machine (e.g. for $M = 200$).

3.4 Visualization and discussion

Figure 1 shows how the optimal β varies with weights w and matching rates g. Higher relative weights in a region lead to a larger range of $\beta(\theta)$ there to make it easier to distinguish those items (e.g., prioritizing the best items induces a β shifted right). Higher relative matching rates $g(\theta)$ have the opposite effect, as frequent sampling naturally increases accuracy for the best items. We formalize this shifting in Appendix Section B.3.

ALGORITHM 1: Nested Bisection

Data: Number of intervals M, match function g
Result: β levels, i.e. $\{t_0 \dots t_{M-1}\}$
Function *main (M, δ, g)*
  while *Uncertainty region for t_{M-2} is bigger than error tolerance* **do**
    Calculate $r(t_{M-2}, 1)$, the rate between current guess for t_{M-2}, and $t_{M-1} = 1$.
    Fixing t_{M-2}, find $t_1 \dots t_{M-3}$ such that $r(t_1,t_2) = r(t_2,t_3) = \cdots = r(t_{M-3},t_{M-2}) = r(t_{M-2},1)$, which can be done through a sequence of bisection routines.
    Calculate $r(0,t_1)$, the rate between current guess for t_1, and $t_0 = 0$.
    Compare $r(t_{M-2},1)$ and $r(0,t_1)$, adjust uncertainty region for t_{M-2} accordingly.
  return $\{t_i\}$

It is interesting that even the basic case, with $w = 1$ and $g = 1$, has a non-trivial β. One would expect, with weight and matching functions that treat all items the same, that β would be linear, i.e. $\beta(\theta) = \theta$. Instead, a third factor non-trivially impacts optimal design: binomial variance is highest near $\beta(\theta) = \frac{1}{2}$. Items that receive positive ratings at such frequency have high-variance scores, and thus the optimal β has a smaller mass of items with such scores.

4 Designing approximately optimal, implementable rating systems

We now turn to our second question: *How does a platform build and implement a real rating system such that buyers behave near-optimally, i.e. according to a calculated β?*. In this section, we give an example design procedure for how a platform would do so, and in the next section we validate our procedure through an experiment on Mechanical Turk.

Recall that $\beta(\theta)$ gives the probability at which an item of quality θ should receive a positive rating. However, the platform cannot force buyers to rate according to this function. Rather, it must ask *questions* of buyers that will induce a proportion $\beta(\theta)$ of them to give positive ratings for an item θ.

Throughout the section, we assume that an optimal

$\beta(\theta)$ has been calculated (for some M, g, and w).

Resources available to platform. We suppose the platform has a set $\mathcal{Y}$ of possible binary questions that it could ask a buyer, e.g., "Are you satisfied with your experience" or "Is this experience your best on our platform?". Informally, at each rating opportunity (i.e., match made), the rater can be shown a single question $y \in \mathcal{Y}$. Let $\psi(\theta, y)$ be the probability an item quality θ would receive a positive rating when the rater is asked question $y \in \mathcal{Y}$.

We further suppose the platform has a set Θ of representative items for which it has access to item quality. Θ, then, is the granularity at which the platform can collect data about historical performance, or otherwise get expert labels. (We assume $M \gg |\Theta|$).

Using known set Θ, the platform can run an experiment to create an estimate $\hat{\psi}(\theta, y), \forall y \in \mathcal{Y}, \theta \in \Theta$.

Design heuristic. How can the platform build an effective rating system using these primitives and β? We consider the following heuristic design: the platform randomly shows a question $y \in \mathcal{Y}$ to each buyer. The choice of the platform is a *distribution* $H(y)$ over $y \in \mathcal{Y}$; in other words, for the platform the design of the system amounts to choosing the frequency with which each question is shown. At each rating opportunity, y is chosen from $\mathcal{Y}$ according to H, independently across opportunities.

Clearly, the probability that an item θ receives a positive rating is: $\tilde{\beta}(\theta) \triangleq \sum_{y \in \mathcal{Y}} \psi(\theta, y) H(y)$.

We want a distribution H such that $\tilde{\beta}(\theta) = \beta(\theta)$ for all $\theta \in [0, 1]$, i.e., that the positive rating probability for each item is exactly the optimal value. However, there may not exist *any* set of questions $\mathcal{Y}$ with associated ψ and choice of H such that $\tilde{\beta} = \beta$.[7]

We propose the following heuristic to address this difficulty. Choose a probability distribution H to minimize the following L_1 distance:

$$\min_{H: \|H\|_1 = 1} \sum_{\theta \in \Theta} |\beta(\theta) - \sum_{y \in \mathcal{Y}} \hat{\psi}(\theta, y) H(y)| \qquad (5)$$

This heuristic uses the data available to the platform, $\hat{\psi}(\theta, y)$ for a set of items $\theta \in \Theta$, and designs H so at least these items receive ratings close to their optimal ratings $\beta(\theta_i)$. Then, as long as ψ is well-behaved, and Θ is representative of the full set, one can hope that $\tilde{\beta}(\theta) \approxeq \beta(\theta)$, for all $\theta \in [0, 1]$.

Discussion. *Real-world analogue & constraints.* A special case of this system is already in place on many platforms, where the same question is always shown. Static systems can be designed by restricting H to only have mass at one question y. More generally, constraints can be used in optimization (5).

Limitations. Our model does not allow for y to be chosen adaptively based on the platform's current knowledge of the item. In practice, this may be a reasonable restriction for implementation purposes. Our model also restricts aggregation to be binary; the platform in our model does not use information on how "hard" a question y is.

5 Mechanical Turk experiment

In the previous sections, we showed how to find an optimal rating function β and we how to apply such a β to design a binary rating system using empirical data. In this section, we deploy an experiment on Amazon Mechanical Turk to apply these insights in practice. First, we collect data that can be used to create a reasonable real-world example of $\psi(\theta, y)$, as a proof of concept with which we can apply our optimization approach. Then, we use this model to demonstrate some key features of optimal and heuristic designs as computed via our methodology, and show that they perform well relative to natural benchmarks. Details of experimental design, simulation methodology, and results are in Appendix A.

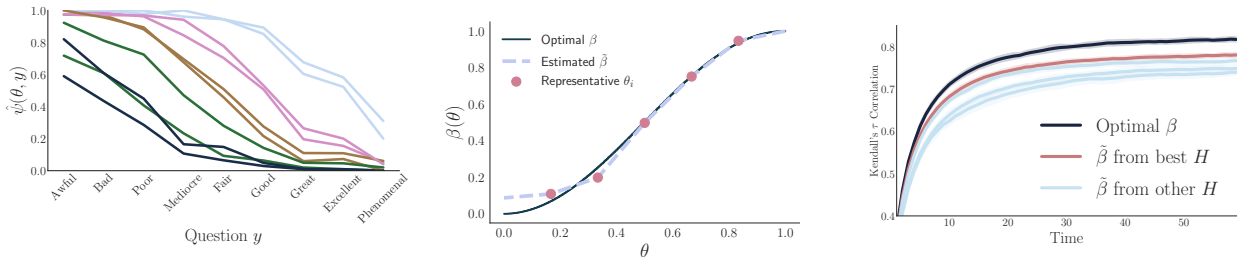
Experiment description. We have a set of 10 English paragraphs of various writing quality, with *expert scores* θ, from a TOEFL book (Educational Testing Service, 2005); there were 5 unique possible experts ratings, i.e. $\theta \in \{.1, .3, .5, .7, .9\}$. For each possible rating, we have two paragraphs who received that score from experts.

We asked workers on Mechanical Turk to rate the writing quality of the paragraphs from a set of adjectives, $\mathcal{Y}$. Using this data, we estimate $\psi(\theta, y)$, i.e., the probability of positive rating when a question based on adjective $y \in \mathcal{Y}$ is shown for paragraph with quality $\theta \in \Theta$. (e.g., "Would you consider this paragraph of quality *[adjective]* or better?") Figure 2a shows our estimated $\hat{\psi}$ for our 10 paragraphs.

Optimization. Next, we find the optimal β for various matching and weight functions using the methods from Section 3. In particular, we have β for all permutations of the cases $g = 1, g = \frac{1+10\theta}{11}$, and $w = 1, w = \theta_1\theta_2(\theta_1 - \theta_2)$, and $w = (1-\theta_1)(1-\theta_2)(\theta_1 - \theta_2)$. Recall that this step does not use experimental data.

Then, using $\hat{\psi}$ and calculated βs, we apply the heuristic from Section 4 to find the distribution H

[7] There are special cases where an exact solution exists. For example, let $\mathcal{Y} = [0, 1]$, and $\psi(\theta, y) = \mathbb{I}[\theta \geq y]$.



(a) Experimental $\hat{\psi}(\theta_i, y)$. Light blue lines are best 2 paragraphs, dark blue the worst.

(b) β vs $\tilde{\beta}$ (using a calculated H) for $w = 1, g = 1$.

(c) Simulated performance for $w = 1, g = 1$.

Figure 2: Experiment and simulation results

with which to sample the questions (adjectives) in $\mathcal{Y}$. Figure 2b shows the optimal β (with $g = 1$, $w = 1$), and estimated $\tilde{\beta}$ from the procedure.

Simulation. Finally, we study the performance of these designs via simulation in various settings. We simulate a system with 500 items and 100 buyers according to the model in Section 3.1, except that matching is *stochastic*: at each time, a random 100 items receive ratings, based on *observed* scores $x_k(\theta)$ rather than true quality θ. Furthermore, in some simulations, we have sellers *entering and exiting* the market with some probability at each time step. We measure the performance of all the designs. For comparison, we also simulate a *naive* $H = \frac{1}{|\mathcal{Y}|}$.

Note that our experiment only provides $\hat{\psi}$ associated with qualities $\theta \in \Theta$, and for simulations we construct a full $\psi(\theta, y), \forall \theta \in [0, 1]$ from these points by averaging and interpolating (in order to model human behavior for the full system). Further, our *calibrated* simulations only provide rough evidence for the approach: although we use real-world data, the simulations assume that our model reflects reality, except for where we deviate as described above.

Results and discussion. Figure 2c shows the simulated performance (as measured by Kendall's τ correlation) of the various designs over time, when $g = 1$. Further plots are in the Appendix Figure 4, showing performance under various weight functions w and matching functions g, and with items entering and exiting the market. We find that:

First, the optimal β for each setting outperforms other possible functions, as expected. The designs are robust to (some) assumptions in the model being broken, especially regarding matching.

Second, the H from our procedure outperforms other designs, but is worse than the optimal system β. In general, the simulated gap between an implemented system and optimal design β provides the platform quantitative insight on the system's sub-optimality.

Third, comparing $\tilde{\beta}$ to β gives qualitative insight on *how* to improve the system. For example, in Figure 2b, $\tilde{\beta}$ is especially inaccurate for $\theta \in [0, .4]$. The platform must thus find better questions for items of such quality. Figure 2a corroborates: our questions cannot separate two low quality paragraphs rated differently by experts (in dark blue and green).

A wide range of recent empirical work has documented that real-world rating systems experience substantial inflation; almost all items receive positive ratings almost every match (Filippas et al., 2017; Fradkin et al., 2017; Tadelis, 2016). Our formulation helps understand how – and whether – such inflation is suboptimal, and provides guidance to platform designers. In particular, rating inflation can be interpreted as a current $\beta(\theta)$ that is very high for almost all item qualities θ. This system is well-performing if the platform objective is to separate the worst items from the rest, or if high quality items receive many more ratings than low quality ones; it is clearly sub-optimal in other cases. Our paper provides a template for how a platform might address such a situation.

Acknowledgements

We thank Michael Bernstein and participants of the Market Design workshop at EC'18. This work was funded in part by the Stanford Cyber Initiative, the National Science Foundation Graduate Research Fellowship grant DGE-114747, and the Office of Naval Research grant N00014-15-1-2786.

References

Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar. Fast and slow learning from reviews. Technical report, National Bureau of Economic Research, 2017.

Omar Besbes and Marco Scarsini. On Information Distortions in Online Ratings. *Operations Research*, 66(3):597–610, June 2018. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.2017.1676.

Gary Bolton, Ben Greiner, and Axel Ockenfels. Engineering trust: reciprocity in the production of reputation information. *Management science*, 59 (2):265–285, 2013.

Lus Cabral and Ali Hortasu. The Dynamics of Seller Reputation: Evidence from Ebay. *The Journal of Industrial Economics*, 58(1):54–78, March 2010. ISSN 1467-6451. doi: 10.1111/j.1467-6451.2010.00405.x.

Yeon-Koo Che and Johannes Horner. Optimal design for social learning. 2015.

James Cook. Uber's internal charts show how its driver-rating system actually works, February 2015. URL `http://www.businessinsider.com/leaked-charts-show-how-ubers\-driver-rating-system-works-2015-2`.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, November 2012. ISBN 978-1-118-58577-1.

J. Pinto da Costa and L. Roque. Limit distribution for the weighted rank correlation coefficient, rw. *REVSTAT-Statistical Journal*, 4(3), 2006.

Yuval Dagan, Yuval Filmus, Ariel Gabizon, and Shay Moran. Twenty (Simple) Questions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 9–21, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4528-6. doi: 10.1145/3055399.3055422.

Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*, volume 38 of *Stochastic Modelling and Applied Probability*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-03310-0 978-3-642-03311-7. DOI: 10.1007/978-3-642-03311-7.

Educational Testing Service. TOEFL iBT Writing Sample Responses, 2005. URL `http://toefl.uobabylon.edu.iq/papers/ibt_2014_12148630.pdf`.

Apostolos Filippas, John J. Horton, and Joseph M. Golden. Reputation in the Long-Run. 2017.

Andrey Fradkin, Elena Grewal, and David Holtz. The Determinants of Online Review Informativeness: Evidence from Field Experiments on Airbnb. 2017. URL `http://andreyfradkin.com/assets/reviews_paper.pdf`.

Snehalkumar (Neil) S. Gaikwad, Mark Whiting, Karolina Ziulkoski, Alipta Ballav, Aaron Gilbee, Senadhipathige S. Niranga, Vibhor Sehgal, Jasmine Lin, Leonardy Kristianto, Angela Richmond-Fuller, Jeff Regino, Durim Morina, Nalin Chhibber, Dinesh Majeti, Sachin Sharma, Kamila Mananova, Dinesh Dhakal, William Dai, Victoria Purynova, Samarth Sandeep, Varshine Chandrakanthan, Tejas Sarma, Adam Ginzberg, Sekandar Matin, Ahmed Nasser, Rohit Nistala, Alexander Stolzoff, Kristy Milland, Vinayak Mathur, Rajan Vaish, Michael S. Bernstein, Catherine Mullings, Shirish Goyal, Dilrukshi Gamage, Christopher Diemert, Mathias Burton, and Sharon Zhou. Boomerang: Rebounding the Consequences of Reputation Feedback on Crowdsourcing Platforms. pages 625–637. ACM Press, 2016. ISBN 978-1-4503-4189-9. doi: 10.1145/2984511.2984542.

Nikhil Garg and Ramesh Johari. Designing Informative Rating Systems for Online Platforms: Evidence from Two Experiments. October 2018. URL `https://arxiv.org/abs/1810.13028v1`.

Peter Glynn and Sandeep Juneja. A large deviations perspective on ordinal optimization. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 1. IEEE, 2004.

Nan Hu, Paul A. Pavlou, and Jie (Jennifer) Zhang. Overcoming the J-Shaped Distribution of Product Reviews. SSRN Scholarly Paper ID 2369332, Social Science Research Network, Rochester, NY, October 2009. URL `https://papers.ssrn.com/abstract=2369332`.

Bar Ifrach, Costis Maglaras, Marco Scarsini, and Anna Zseleva. Bayesian Social Learning from Consumer Reviews. SSRN Scholarly Paper ID 2293158, Social Science Research Network, Rochester, NY, December 2017. URL `https://papers.ssrn.com/abstract=2293158`.

Nicole Immorlica, Brendan Lucier, Brian Rogers, and others. Emergence of cooperation in anonymous social networks through social capital. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.

Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching While Learning. In *Proceedings of the 2017 ACM Conference on Economics and*

Computation, EC '17, pages 119–119, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4527-9. doi: 10.1145/3033274.3084095. event-place: Cambridge, Massachusetts, USA.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvri, and Zheng Wen. DCM Bandits: Learning to Rank with Multiple Clicks. *arXiv:1602.03146 [cs, stat]*, February 2016. URL http://arxiv.org/abs/1602.03146. arXiv: 1602.03146.

Francis Maes, Louis Wehenkel, and Damien Ernst. Automatic Discovery of Ranking Formulas for Playing with Multi-armed Bandits. In *Recent Advances in Reinforcement Learning*, Lecture Notes in Computer Science, pages 5–17. Springer, Berlin, Heidelberg, September 2011. ISBN 978-3-642-29945-2 978-3-642-29946-9. doi: 10.1007/978-3-642-29946-9_5. URL https://link.springer.com/chapter/10.1007/978-3-642-29946-9_5.

Chris Nosko and Steven Tadelis. The Limits of Reputation in Platform Markets: An Empirical Analysis and Field Experiment. Working Paper 20830, National Bureau of Economic Research, January 2015. URL http://www.nber.org/papers/w20830.

Yiangos Papanastasiou, Kostas Bimpikis, and Nicos Savva. Crowdsourcing Exploration. *Management Science*, 64(4):1727–1746, April 2017. ISSN 0025-1909. doi: 10.1287/mnsc.2016.2697.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM, 2008.

Shiva Rajaraman. Five Stars Dominate Ratings, September 2009. URL https://youtube.googleblog.com/2009/09/five-stars-dominate-ratings.html.

Steven Tadelis. Reputation and feedback systems in online platform markets. *Annual Review of Economics*, 8:321–340, 2016.

Agostino Tarsitano. Comparing the effectiveness of rank correlation statistics. *Working Papers, Universita della Calabria, Dipartimento di Economia e Statistica*, pages 1–25, 2009.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.

Georgios Zervas, Davide Proserpio, and John Byers. A First Look at Online Reputation on Airbnb, Where Every Stay is Above Average. SSRN Scholarly Paper ID 2554500, Social Science Research Network, Rochester, NY, January 2015. URL http://papers.ssrn.com/abstract=2554500.