Multiagent Evaluation Mechanisms

Tal Alon
Technion, Israel

Magdalen Dobson
Carnegie Mellon University, USA

Ariel D. Procaccia
Carnegie Mellon University, USA

Inbal Talgam-Cohen
Technion, Israel

Jamie Tucker-Foltz
University of Cambridge, UK

Abstract

We consider settings where agents are evaluated based on observed features, and assume they seek to achieve feature values that bring about good evaluations. Our goal is to craft evaluation mechanisms that incentivize the agents to invest effort in desirable actions; a notable application is the design of course grading schemes. Previous work has studied this problem in the case of a single agent. By contrast, we investigate the general, multi-agent model, and provide a complete characterization of its computational complexity.

1 Introduction

Any reader who has ever taught a course would have undoubtedly faced some variant of the grading-scheme dilemma: Should the final exam count for 30% of the grade, and the homework assignments for 40%? Or should these two components perhaps be weighted equally? Should the lowest homework grade be dropped? Admittedly, grades only serve as a proxy for students' (unobservable) learning outcomes. But once a grading scheme is in place, students will optimize their grades by investing effort accordingly. Therefore, the grading scheme must be designed to encourage desirable behaviors. For example, a participation grade may make some students come to class, but those same students — who are now short on time — may elect to cheat on their homework assignments.

The design of course grading schemes is an instance of a much broader challenge. Whenever an *evaluator* designs a scheme for evaluating an *agent* based on observed features, the agent is incentivized to achieve feature values that lead to a good evaluation. The hope is that the agent will do so through genuine self-improvement rather than blatant gaming. The evaluation of creditworthiness through credit history in the United States serves as an especially egregious example: instead of promoting true financial responsibility, it encourages idiosyncratic practices such as using a specific percentage of one's credit card limit.

In a very recent paper, Kleinberg and Raghavan (2019) model and analyze these scenarios. In their model, an agent has a given amount of *effort* that can be invested in different *actions* (e.g., attempt to solve a homework assignment or cheat). There are also *effort conversion* functions that map the levels of effort invested in each action to *features* (e.g., homework grade, exam grade, or participation grade). The evaluator's task is to design a *mechanism* that maps the feature values to a score, which coincides with the agent's *utility*. The agent seeks to distribute effort between actions to achieve maximum utility. The evaluator's goal, then, is to design the mechanism to elicit a desirable *effort profile*.

Importantly, an instance of the evaluation problem of Kleinberg and Raghavan (2019) includes only a single agent — or, equivalently, multiple agents that share the same model. However, in the domains of interest there are multiple *types* of agents. For example, one student might optimize exam grades by studying alone, whereas another would derive more benefit from studying with peers. With this in mind, we wish to extend the model and results of Kleinberg and Raghavan to the multi-agent case. In other words, our main research challenge is this:

> *Given a set of different agent models, design mechanisms that induce all agents, or as many agents as possible, to invest effort in desirable actions.*

1.1 Our Results

The foregoing challenge gives rise to multiple problems, each of which is determined by the answers to the following three questions.

First, what requirements must the mechanism satisfy? The minimal assumption made by Kleinberg and Raghavan is that the mechanism must be *monotone*, meaning that agents should receive higher payoffs for higher feature scores. A more restrictive requirement is that the mechanism be *linear*, meaning that the payoff is just a linear combination of all feature scores.

Second, what is the goal of the evaluator? Is there some specific *admissible profile* of effort investment the evaluator wishes to incentivize, or will they be content with any profile, as long as all of the actions which the agent invests a nonzero amount of effort into are *admissible actions*?

Third, are we interested in incentivizing *all agents* in a particular way, or just a *maximum number of agents*?

With only one agent, the main result of Kleinberg and Raghavan is that the answers to these questions do not matter: whenever a monotone mechanism exists a linear mechanism exists, and whether an effort profile can be incentivized

Problem #	Type of mechanism	Incentivize	Admissible set	Complexity
1	monotone	all agents	admissible profile	P
2	monotone	all agents	admissible actions	P (const. n) NP-c (gen.)
3	monotone	max # of agents	admissible profile	NP-c
4	monotone	max # of agents	admissible actions	NP-c
5	linear	all agents	admissible profile	P
6	linear	all agents	admissible actions	P (const. n) NP-c (gen.)
7	linear	max # of agents	admissible profile	P (const. n) NP-c (gen.)
8	linear	max # of agents	admissible actions	P (const. n) NP-c (gen.)

Table 1: Complexity of the 8 different variants of the evaluation problem. Note that n is the number of features, and "NP-c" stands for NP-complete.

depends only on the actions it is supported by. However, with multiple agents, we find striking differences between the problem variants, both qualitatively and in terms of computational complexity. We provide a complete classification of the complexity of each of these 8 problems, as shown in Table 1. For each problem, we also consider the complexity for the realistic restriction of a constant number of features n. (For example, even in MOOCs with massively-many students, the number of features factored into the final grade is likely to be held constant.) Problems 1-4 are analyzed in Section 3, and Problems 5-8 are analyzed in Section 4.

1.2 Related Work

There are two main lines of related work. First, evaluation can be viewed as *classifying* strategic agents (into classes such as "A students", "B students", etc.). Self-interested agents facing classification may invest in distorting their true attributes, in order to steer the classifier away from their "ground-truth" class. The goal in *strategic classification* is to build classifiers robust to such *gaming* (Hardt et al. 2016). Our goal is in some sense opposite to this line of work — we aim to *encourage* agents to invest in changing their features, but by choosing desirable actions like studying over undesirable ones like cheating. In other words, in our case the evaluation is not meant to expose some ground truth, but rather to incentivize worthwhile behavior. Strategic classification is part of a more general literature on learning in the presence of strategic behavior (Meir, Procaccia, and Rosenschein 2008; 2012; Dekel, Fischer, and Procaccia 2010).

A second line of research closely related to our work is *contract design*, a branch of microeconomics (Grossman and Hart 1983) that has recently gained interest in computer science (Babaioff, Feldman, and Nisan 2006; Dütting, Roughgarden, and Talgam-Cohen 2019). The precise relation between our model and the classic principal-agent, hidden-action[1] model from microeconomics is explained in Appendix A.[2] In a nutshell, the basic setting of our model can be reinterpreted as a simplified principal-agent one, in which the principal (the evaluator in our model) has no inherent interest in the agents' outputs except to incentivize the agents to choose permissible hidden actions. Given the connection between the models, to avoid confusion it is important to note here that we use the term *linear mechanism* for an entirely different object than the *linear contract* term that is standard in microeconomics — see the appendix for details.[3] We also diverge from previous work on contract design for multiple agents in our motivation for applying a *unified* approach to incentivizing the agents, instead of dealing with each of them separately: rather than aiming to encourage cooperation or optimize information as is common in the contract design literature, we are motivated by the fairness requirement that all agents face a single uniform evaluation mechanism (see the appendix for more details).

In parallel work, Xiao et al. (2020) also study the problem of incentivizing multiple agents under a single mechanism. In their model, actions are directly observable, and in designing the contract, the principal is motivated by profit and has to compensate the agents at personal expense. Hence, their model applies to an entirely different set of principal-agent problems than ours.

2 The Model

For consistency we adopt the notation of Kleinberg and Raghavan (2019) where possible. An instance of the *evaluation problem* consists of actions $1, 2, \ldots, m$ (indexed by j); features $F_1,F_2,\ldots,F_n$ (indexed by i); and agents (e.g., students) $S = \{s_1,s_2,\ldots,s_\ell\}$ (indexed by k). Each agent s_k has a matrix α^k in $\mathbb{R}^{m\times n}_{\geq 0}$ called their *effort conversion matrix*. Entry $\alpha^k_{j,i} \in \mathbb{R}_{\geq 0}$ (which we assume is described using a polynomial number of bits in m, n) specifies how effort put into action j translates into feature i (as specified in the next paragraph). We assume that every agent s_k has the ability to affect every feature, that is, no matrix α^k has an all-zero column.[4] We denote the j^{th} row and its entries by $\alpha^k_j = (\alpha^k_{j,1}, \ldots, \alpha^k_{j,n})$. It is often convenient to describe an instance of the evaluation problem as an *effort graph* as depicted in Figure 1. The instance in Figure 1 has $m = 2$ actions, $n = 2$ features and $\ell = 2$ agents, and the conversion matrices are $\alpha^1_1 = (4, p), \alpha^1_2 = (0, 9)$ for the first agent and $\alpha^2_1 = (p, 4), \alpha^2_2 = (9, 0)$ for the second.

Each agent s_k has a budget of one unit of effort to divide

[1] As opposed to hidden-type models; note that while we deal with different types of agents, these are not hidden.

[2] The appendix is included in the full version of our paper, available at `http://procaccia.info`.

[3] We use the term linear mechanism to be consistent with (Kleinberg and Raghavan 2019).

[4] This is implicitly assumed in (Kleinberg and Raghavan 2019).

among different actions.[5] Their choice of how to divide their budget is specified by their *effort profile* x^k, where $x_j^k \geq 0$ (or x_j — we sometimes omit the k index where clear from context) is the effort they invest in action j, and $\sum_j x_j^k \leq 1$ (*feasibility*). We refer to the set of all feasible effort profiles as $\mathcal{X}$.

An effort profile is converted into the agent's n features as follows: $F^k(x^k)_i = \sum_j \alpha_{j,i}^k x_j^k$ for every $i \in [n]$. In words, for feature F_i, the effort s_k puts into action j is multiplied by $\alpha_{j,i}^k$, and this is summed over all actions. Note that Kleinberg and Raghavan (2019) introduce a generalization: they define $F^k(x^k)_i = f_i^k(\sum_j \alpha_{j,i}^k x_j^k)$, where f_i^k is a concave, strictly increasing function. We use the simpler form for ease of exposition, and indeed most of our results hold for the more general model — see Appendix I for details.

An *evaluation mechanism* is a function $M : \mathbb{R}_{\geq 0}^n \to \mathbb{R}_{\geq 0}$ that maps an agent's features to a score or *payoff*. The score coincides with the agent's utility. Given a mechanism, the agent chooses an effort profile x^k that maximizes their score; we then say x^k is *incentivized* by the mechanism. The design goal is to have the mechanism incentivize all agents, or as many agents as possible, to invest only in a prescribed set of *admissible* profiles (we assume that if several profiles are incentivized, the agent breaks ties in favor of admissible ones). We use $\mathcal{A} \subseteq \mathcal{X}$ to denote the set of admissible profiles, and consider two different problem variants depending on the form of $\mathcal{A}$: (1) In the *admissible profile* variant, $|\mathcal{A}| = 1$, meaning that the agents must be incentivized to choose a particular effort profile, which is given as part of the input. (2) In the *admissible actions* variant, $\mathcal{A}$ is implicitly specified by a subset of actions $A \subseteq [m]$, and an effort profile is admissible if and only if it is supported only over admissible actions ($x_j^k = 0$ for every $j \notin A$).

We consider two main classes of evaluation mechanisms: monotone mechanisms and their subclass of linear mechanisms. An evaluation mechanism is *monotone* if two conditions hold: (i) for every two feature vectors $F' \geq F$, it holds that $M(F') \geq M(F)$;[6] and (ii) for every feature vector F there exists a subset S of features such that increasing all features F_S in the subset strictly increases $M(F)$.[7] An evaluation mechanism is *linear* if $M(F)$ is a multilinear function in the features, namely, $M(F) = \sum_i \beta_i F_i$ where $\beta_i \geq 0$ for every i and $\beta_{i'} > 0$ for some i'.

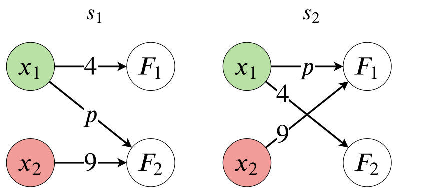

Figure 1: A two-agent instance of the evaluation problem, where $p \in [1, 8]$ is an arbitrary parameter.

2.1 Examples

The following examples illustrate the complexity that is added to the evaluation problem when there are multiple agents.

Example 2.1. Returning to the classroom setting, suppose that there are two types of students, s_1 and s_2, who can choose between studying (action 1) and cheating (action 2). Studying improves both test scores (feature F_1) and homework scores (feature F_2) for both types, while cheating improves just one of the scores by an even greater amount. The effort conversion rates in such a scenario might be as depicted in Figure 1, where $1 \leq p \leq 8$.

Since cheating only improves one kind of score, there are simple linear mechanisms that can incentivize studying for either student type in isolation, no matter what p is: for s_1, set $\beta := (1, 0)$ (final score depends only on the test), and for s_2, set $\beta := (0, 1)$ (final score depends only on the homework). But what if we wish to simultaneously incentivize both student types to study?

At $p = 6$, there is a still a linear mechanism that works. Taking $\beta := (1, 1)$, the marginal benefit toward studying is 10, while the marginal benefit toward cheating is 9, so both student types will invest all of their effort into studying.

At $p = 4$, no linear mechanism exists. For if some $\beta = (\beta_1, \beta_2)$ incentivizes s_1 to study, we must have $4\beta_1 + 4\beta_2 \geq 9\beta_2$, or in other words, $4\beta_1 \geq 5\beta_2$. Analogously, if that same β incentivizes s_2 to study, we must have $4\beta_2 \geq 5\beta_1$. This is only satisfied by $\beta = (0, 0)$, which violates the monotonicity requirement that at least one coordinate be strictly positive (and makes no sense as a classroom scoring method). Thus, no linear mechanism can simultaneously incentivize both student types to study. However, consider a *nonlinear*, monotone mechanism: $M(F_1, F_2) := \min(F_1, F_2)$. Neither type of student is incentivized to cheat under this mechanism, since that will not improve their minimum score.

At $p = 1$, there is no monotone mechanism at all, not even a nonlinear one. Supposing there was such an $M : \mathbb{R}_{\geq 0}^2 \to \mathbb{R}_{\geq 0}$, consider the choice of an s_1 student between two different profiles: the admissible profile $(1, 0)$, and the inadmissible profile $(\frac{1}{2}, \frac{1}{2})$. If s_1 chooses the admissible profile, they obtain a feature vector $(4, 1)$, and if they choose the inadmissible profile, they obtain the feature vector $(2, 5)$. Since we are assuming M incentivizes only studying, we must therefore have $M(4, 1) \geq M(2, 5)$. Monotonicity implies $M(2, 5) > M(1, 4)$, so $M(4, 1) > M(1, 4)$. However, by a completely symmetric argument, for s_2 students to be incen-

[5] In Kleinberg and Raghavan (2019), the agent has an arbitrary effort budget B. Note it is without loss of generality to assume $B = 1$ for all agents as we do, since any discrepancies in effort budgets can instead be realized by scaling the effort conversion matrices.

[6] Throughout the paper, whenever we write an inequality between two vectors, it means that the inequality holds in each coordinate.

[7] Together with the assumption that no effort conversion matrix has a column of zeros, condition (ii) implies there is always *potential* to increase the score by investing more effort, so all agents are strictly incentivized to exhaust their effort budgets. Without condition (ii), the evaluation problem would be trivial, since we could always just use the mechanism that gives a payoff of zero no matter what the feature scores are.

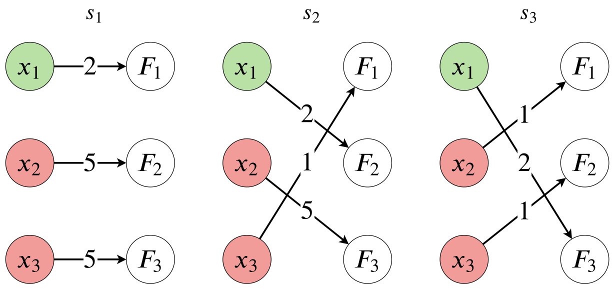

Figure 2: A three-agent instance of the evaluation problem exhibiting the power of nonlinear mechanisms. This construction can be generalized to any number of agents.

tivized to study we must have $M(1,4) > M(4,1)$, which is a contradiction. Thus, no monotone mechanism can exist.

One of the most remarkable conclusions from Example 2.1 is that, in stark contrast to the one-agent case, nonlinear mechanisms can succeed where linear mechanisms fail. One can interpret the scenarios where nonlinear mechanisms gain an advantage from a machine learning perspective. Nonlinear classifiers can see more complex relationships among data, such as the conjunction of two separate conditions like, “in order to be in the positive class, feature 1 must have value at least x *and* feature 2 must have value at least y.” Analogously, a nonlinear evaluation mechanism can make more complicated distinctions between desirable and undesirable behavior, such as, “in order to get a high score, feature 1 must have value at least x *and* feature 2 must have value at least y.” Nonlinear mechanisms are necessary when the effort conversion rates of several agents combine to form a complex boundary in the feature space between the results of desirable agent behavior and undesirable agent behavior, which cannot be linearly separated. This effect can be quite dramatic: as the following extreme example shows, there exist situations where the best linear mechanisms perform arbitrarily worse than the best nonlinear ones.

Example 2.2. For any positive integer n, define an instance of the evaluation problem with n actions, n features, and n agents, where only action 1 is admissible, and the rate of effort conversion for agent s_k from action j to feature F_i is

$$\alpha_{ji}^{k} := \begin{cases} 1 & \text{if } i < k \\ 2 & \text{if } i = k \\ 5 & \text{if } i > k \end{cases}$$

if $j+k-1 = i \mod n$, and zero otherwise. Figure 2 shows an example of this construction when $n = 3$.

Suppose some pair of agents $s_k, s_{k'}$ where $k' > k$ could be jointly incentivized to invest only in action 1 with some linear mechanism β. For s_k to be incentivized to invest only in action 1, we must have $2\beta_k \geq 5\beta_{k'}$, and for $s_{k'}$ we must have $2\beta_{k'} \geq \beta_k$. Therefore, $\beta_k \geq \frac{5}{4}\beta_k$, so $\beta_k = 0$, in which case monotonicity implies it is strictly preferable for s_k to invest in some other, inadmissible action yielding a nonzero payoff. Hence, no linear mechanism can incentivize more than one agent to invest only in action 1. However, we will see in Section 3.1 that *all* agents can be incentivized to invest only in action 1 with a monotone, nonlinear mechanism.

3 Monotone Mechanisms

In this section we first describe one of our main contributions — a useful characterization of when it is possible to jointly incentivize a given set of agents to choose admissible actions via a monotone mechanism. The proof is constructive, giving an efficiently computable $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ with the guarantee that if any monotone mechanism “works,” so does M. Using this characterization, we present polynomial-time algorithms to solve Problem 1 for an unbounded number of features, and Problem 2 for a constant number of features. We then present hardness results for the remaining problems in the top half of Table 1.

3.1 Imitation Graphs

The central obstacle in the multi-agent evaluation problem is that one agent may be able to achieve good scores by choosing admissible actions, while another agent may be able to achieve even better scores by choosing inadmissible actions. In such scenarios, we say that the second agent is able to *imitate* the first one. The key observation is that, when this happens, the second agent must be given a greater payoff than the first agent, rewarding them for not acting in this undesirable way. This idea will allow us to characterize exactly when it is possible to jointly incentivize multiple agent types. Moreover it will imply that the incentivizing mechanism will have quite a natural form, ranking agents by their capability to emulate others’ achievements, and assigning them payoffs according to this ranking.

To formally define imitation, we shall refer to two agents s_1 and s_2, where we somewhat abuse notation using s_1, s_2 for arbitrary agents as opposed to the agents with index $k = 1, 2$. We use this convention throughout Section 3 to avoid excessive subscripts.

Fix an admissible action profile $x^*(s)$ for each agent $s \in S$. We say that agent s_1 can *imitate* agent s_2 with respect to x^* if s_1 can play an inadmissible action profile x such that $F^{s_1}(x) \geq F^{s_2}(x^*(s_2))$. If x can be chosen so that $F^{s_1}(x) > F^{s_2}(x^*(s_2))$, we say that s_1 can *strictly* imitate s_2. The *imitation graph* with respect to x^* is the directed graph with vertex set S and an edge from s_1 to s_2 if and only if s_1 can imitate s_2. If s_1 can strictly imitate s_2, we say that (s_1, s_2) is a *strict edge*.

Theorem 3.1. *It is possible to incentivize all agents in S to choose effort profiles in $\mathcal{A}$ using a monotone mechanism if and only if there exists some $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges.*

Proof. For the forward direction, suppose $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a monotone mechanism that incentivizes all agents to invest only in admissible actions. Then for each $s \in S$, choose $x^*(s)$ to be any admissible best response of s under M. Suppose toward a contradiction that the imitation graph with respect to x^* contains a directed cycle s_1,$s_2, \ldots$,s_q, s_1, where

(s_q, s_1) is a strict edge. Then for each k from 1 to $q-1$, we must have

$$M(F^{s_k}(x^*(s_k))) \geq M(F^{s_{k+1}}(x^*(s_{k+1}))),$$

for otherwise, if $M(F^{s_{k+1}}(x^*(s_{k+1}))) > M(F^{s_k}(x^*(s_k)))$ then agent s_k could deviate from $x^*(s_k)$ and choose some inadmissible action x such that

$$F^{s_k}(x) \geq F^{s_{k+1}}(x^*(s_{k+1})),$$

receiving a strictly greater payoff:

$$M(F^{s_k}(x)) \geq M(F^{s_{k+1}}(x^*(s_{k+1}))) > M(F^{s_k}(x^*(s_k)))$$

(here the first inequality follows from monotonicity condition (i)). Additionally, we must have

$$M(F^{s_q}(x^*(s_q))) > M(F^{s_1}(x^*(s_1))),$$

for otherwise, if $M(F^{s_1}(x^*(s_1))) \geq M(F^{s_q}(x^*(s_q)))$, agent s_q could deviate from $x^*(s_q)$ and choose some inadmissible action x such that $F^{s_q}(x) > F^{s_1}(x^*(s_1))$, receiving a strictly greater payoff:

$$M(F^{s_q}(x)) > M(F^{s_1}(x^*(s_1))) \geq M(F^{s_q}(x^*(s_q)))$$

(here the first inequality follows from monotonicity condition (ii)). Thus, we have an inconsistent cycle of inequalities

$$\begin{aligned} M(F^{s_1}(x^*(s_1))) &\geq M(F^{s_2}(x^*(s_2))) \geq \dots \\ &\geq M(F^{s_q}(x^*(s_q))) > M(F^{s_1}(x^*(s_1))). \end{aligned}$$

We have reached a contradiction, so it must be that there are no cycles containing any strict edges.

For the backward direction, let G be the imitation graph with respect to some x^*, and assume that G has no directed cycles containing any strict edges. We topologically sort the strongly connected components of G in decreasing order, and let $v : S \to \{1, 2, \dots, |S|\}$ give the index of each vertex's component in the topological sort (v already provides a rough ranking of the agents). Let $m : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the function that takes the minimum value of all coordinates in a vector, and let $B \in \mathbb{R}_{>0}$ be a strict upper bound on $m(F)$ for any feature vector F that is attainable by any agent in S. Consider the mechanism

$$\begin{aligned} M(F) := \max \Big\{ v(s) + \frac{m(F - F^s(x^*(s)))}{B} \;\Big|\; \\ s \in S,\ F \geq F^s(x^*(s)) \Big\} \end{aligned} \tag{1}$$

It is not hard to verify that M satisfies both conditions for monotonicity.[8] We claim that M incentivizes every $s \in S$ to play the admissible action $x^*(s)$.

Suppose, toward a contradiction, that for some agent s_1, there existed some alternative, inadmissible effort profile x yielding a strictly higher payoff, i.e.,

$$M(F^{s_1}(x)) > M(F^{s_1}(x^*(s_1))).$$

By the definition of M, this means that

$$\begin{aligned} &\max \Big\{ v(s_2) + \frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} \;\Big|\; \\ &\qquad s_2 \in S,\ F^{s_1}(x) \geq F^{s_2}(x^*(s_2)) \Big\} \\ &> \max \Big\{ v(s_3) + \frac{m(F^{s_1}(x^*(s_1)) - F^{s_3}(x^*(s_3)))}{B} \;\Big|\; \\ &\qquad s_3 \in S,\ F^{s_1}(x^*(s_1)) \geq F^{s_3}(x^*(s_3)) \Big\}. \end{aligned}$$

Taking any s_2 on the LHS that realizes the maximum, and plugging in s_1 for s_3 on the RHS, this becomes

$$v(s_2) + \frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} > v(s_1).$$

Since $m(F) < B$ for any attainable feature vector F, it follows that $v(s_2) + 1 > v(s_1)$. Since v is integer-valued, this means $v(s_2) \geq v(s_1)$.

On the other hand, $F^{s_1}(x) \geq F^{s_2}(x^*(s_2))$ implies $(s_1, s_2) \in E(G)$, so $v(s_1) \geq v(s_2)$. Thus, we have $v(s_1) = v(s_2)$, meaning that s_1 and s_2 are in the same strongly connected component of G. Also,

$$\frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} > v(s_1) - v(s_2) = 0,$$

which implies $F^{s_1}(x) > F^{s_2}(x^*(s_2))$, so (s_1, s_2) is a strict edge. But it is impossible to have a strict edge between two vertices in the same strongly connected component, as this would imply that G has a cycle containing that strict edge, contradicting our hypothesis. We have a contradiction, so M incentivizes all agents to play according to x^*. □

Notice that the imitation graph in Example 2.2 with respect to all agents investing all effort in action 1 consists of a strict edge $(s_{k'}, s_k)$ whenever $k' > k$. Since this graph has no cycles, Theorem 3.1 implies there is a monotone mechanism that incentivizes all agents to invest only in action 1, in particular the mechanism specified in (1). Ignoring the small payoff summand that is a fraction over B (which is only necessary for satisfying condition (ii) of monotonicity), the payoff of this mechanism is

$$M(F) \approx \max_{i \in [n]} \{ n - i + 1 \mid F_i \geq 2 \}.$$

In words, all agents are incentivized to focus all of their effort on raising the feature of smallest index in which they can score at least 2. For each agent, this feature is always the one with an edge from x_1 in the effort graph (see Figure 2), so all agents will invest only in action 1.

3.2 Incentivizing All Agents

Theorem 3.1 directly leads to a simple algorithm to solve Problem 1.

Corollary 3.2. *There is a polynomial-time algorithm to find a monotone mechanism that incentivizes all agents to choose a specific effort profile, or determine that no such mechanism exists.*

[8] Technically speaking, the mechanism is undefined for off-equilibrium-path strategies of scoring lower than any agent should ever score. This can be fixed by adding a dummy agent with an effort conversion rate of zero from every action to every feature.

Proof. Since $|\mathcal{A}| = 1$, there is only one possible $x^* : S \to \mathcal{A}$ to choose from, and thus only one possible imitation graph G. For each strict edge $(s_1, s_2) \in E(G)$, we just check to see if there is a path in G from s_2 to s_1. By Theorem 3.1, it is possible to jointly incentivize all agents if and only if none of these paths exist.

The only potential difficulty lies in constructing this imitation graph in the first place. We show in Lemma B.1 of Appendix B that this can be accomplished in polynomial time using linear programming. □

Solving Problem 2 is trickier, since to use our characterization we must search for an assignment $x^* : S \to \mathcal{A}$ prescribing which admissible action profile we would like each agent to choose. While at first glance this might appear hopelessly intractable, we make an observation that turns out to help when the number of features is constant: If there exists any $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges, then there exist profiles for some nonempty subset of agents $T \subseteq S$ such that

1. no agents in $S \setminus T$ can imitate any agents in T, and
2. no agents in T can strictly imitate any agents in T.

Informally, the observation follows from topologically sorting the strongly connected components of the imitation graph, and noticing that the first component must have these two properties. Given the observation, if such a subset of agents T and their effort profiles can be found in polynomial time, those agents can be removed, since they can no longer create a cycle with any of the remaining agents. If it is then possible to keep removing sets of agents in this manner, then the final imitation graph will have no cycles containing strict edges; otherwise, we can conclude impossibility for the given problem instance.

Finding a subset T and corresponding profiles can be achieved using an iterative marking algorithm, formally presented in Appendix C.1. However, it relies on the ability to efficiently answer the simple question, "Is there some admissible profile that s_1 can play that no agent in some given set R can (strictly) imitate?" Formally, this predicate is,

$$\exists x^1 \in \mathcal{A},\ \forall s_2 \in R,$$

$$\neg\left(\exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i \geq F^{s_1}(x^1)_i\right)$$

(for the strict version, we have a strict inequality). It turns out that this is computable in polynomial time for a constant number of features n, but is NP-hard in general, and so is Problem 2 (see Appendix C for the details).

Theorem 3.3. *The problem of finding a monotone mechanism that incentivizes all agents to choose admissible actions, or determining that no such mechanism exists, is*

1. *solvable in polynomial time for a constant number of features n, and*
2. *NP-complete for unbounded n.*

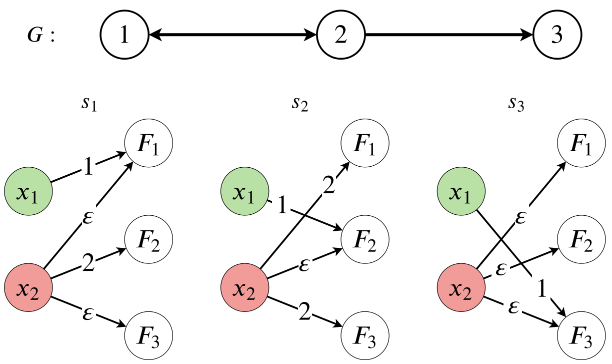

Figure 3: An input graph G and the corresponding evaluation problem produced by the reduction.

3.3 Incentivizing a Maximum Number of Agents

Once the imitation graph for all agents has been constructed, Theorem 3.1 implies that we can incentivize any subset of agents whose induced subgraph has no cycles with strict edges. When all edges are strict, this is an instance of the NP-complete Feedback Vertex Set problem: given a graph, it asks for a minimum-size subset of vertices whose deletion would eliminate all cycles. It turns out that there is a reduction in the other direction too, since all directed graphs can be constructed as imitation graphs — even with just two features! This proves that Problems 3 and 4 are NP-complete.

Theorem 3.4. *The problems of finding a monotone mechanism that incentivizes a maximum number of agents to choose admissible actions / a specific admissible profile are NP-complete even for a constant number of features.*

We will sketch the proof of NP-hardness for an *unbounded* number of features, leaving the more complicated reduction with only two features for Appendix E. Suppose we are given an instance of Feedback Vertex Set, that is, a graph G with n vertices. For convenience, assume $V(G) = [n]$. We construct an instance of the evaluation problem with agents s_1,s_2, . . . ,s_n, features F_1,F_2, . . . ,F_n, and 2 actions, where action 1 is admissible and action 2 is inadmissible. For each $i, j \in [n]$, define

$$\alpha^k_{1,i} := \begin{cases} 1 & i = k \\ 0 & \text{otherwise} \end{cases}, \quad \alpha^k_{2,i} := \begin{cases} 2 & (k,i) \in E(G) \\ \varepsilon & \text{otherwise} \end{cases}$$

(see Figure 3 for an example where $n = 3$).

It is proved in Appendix D that, for $0 < \varepsilon < 1$, G is the imitation graph with respect to the profile assignment of $(1, 0)$ for all agents, and all edges are strict edges. By Theorem 3.1, G has a feedback vertex set of size at most q if and only if at least $n - q$ agents (namely, those not in the feedback vertex set) can be jointly incentivized to invest only in action 1.

4 Linear Mechanisms

While nonlinear monotone mechanisms can incentivize arbitrarily more agents than linear ones (see Example 2.2), there are still many reasons to consider the problem of finding a linear mechanism to incentivize multiple agents. For one, what we call a linear mechanism coincides with traditional contracts investigated in contract theory (see Appendix A). Linear mechanisms also bear more similarity to the kinds of grading schemes commonly used in practice. Furthermore, one might hope that, since linear mechanisms are simpler than monotone mechanisms, finding linear mechanisms might be an easier problem. This intuition turns out to be partially correct: under the very reasonable assumption that the number of features is held constant, each problem we consider in this section has a polynomial time algorithm, although some are hard in general.

4.1 Algorithmic Results

Observe first that, when responding to a linear mechanism, agents can simply compute the marginal payoff toward each of their actions, and invest effort only in the most profitable ones. Therefore, an agent may only split their effort among multiple actions if they are all tied for the highest marginal payoff. If we only care about an agent investing in any one of a set of multiple admissible actions, we need only ensure that one of those admissible actions gives the highest marginal payoff.

Motivated by these observations, we introduce the following notation: when, for a given agent s_k, some action j_1 yields a weakly greater marginal payoff under a linear mechanism $\beta \in \mathbb{R}^n$ than some other action j_2, we say that β satisfies the constraint $h(k, j_1, j_2)$. Since each $h(k, j_1, j_2)$ can be written as a linear constraint over the space of linear mechanisms $\mathbb{R}^n$, we immediately have an algorithm for Problem 5.

Theorem 4.1. *There is a polynomial-time algorithm to find a linear mechanism that incentivizes all agents to choose a specific effort profile, or determine that no such mechanism exists.*

Proof. To solve this problem, we must determine if there exists $\beta \in \mathbb{R}^n$ such that, for every agent $s_k \in S$ and every action j_1 in the support of the admissible profile, for every alternative action $j_2 \in [m]$ the constraint $h(k, j_1, j_2)$ is satisfied (i.e., $\alpha^k_{j_1} \cdot \beta \geq \alpha^k_{j_2} \cdot \beta$). This reduces to testing the feasibility of a linear program with n variables and at most ℓm^2 constraints (were ℓ is the number of agents), which is solvable in polynomial time. □

Jumping to Problem 7, we do not require that β satisfy these constraints for all $k \in [\ell]$, just for as many k as possible. Let $\mathcal{L}_k$ be the polytope in $\mathbb{R}^n$ consisting of all points β that satisfy $h(k, j_1, j_2)$ for all actions j_1 in the support of the admissible profile, and for all actions $j_2 \in [m]$. Our objective is then to find a point in the intersection of a maximum number of the $\mathcal{L}_k$ polytopes. This is no longer a convex optimization problem like Problem 5. Yet we can solve it efficiently when n is a constant, using a geometric data structure known as a *hyperplane arrangement* (Goodman and O'Rourke 1997, Chapter 28).

An arrangement decomposes $\mathbb{R}^n$ into connected open *cells*, where each cell is a maximal connected region in the intersection of a subset of the hyperplanes that is not intersected by any other hyperplane. The key property that we will use is that all points within a given cell are equivalent in terms of which of the linear constraints they satisfy. This implies that, to test whether a given predicate on the constraints holds for any point in $\mathbb{R}^n$, it suffices to check only one point from each cell. This is tractable when n is constant, since it is known that the arrangement of p hyperplanes decomposes $\mathbb{R}^n$ into $O(p^n)$ cells, and that the arrangement can be computed in $O(p^n)$ time.

Algorithm 1: An algorithm for Problem 7.

```
Input: An instance of the evaluation problem with a
       single admissible profile $x^*$
Output: A linear mechanism $\beta$ that incentivizes a
        maximum number of agents to invest effort
        according to $x^*$
1  $\mathcal{R} \leftarrow$ arrangement of all hyperplanes for constraints
     $h(k, j_1, j_2)$ for all $k \in [\ell]$, $j_1 \in \mathcal{S}(x^*)$, and
     $j_2 \in [m]$;
2  max $\leftarrow -1$;
3  for each cell $C \in \mathcal{R}$ do
4      $\beta' \leftarrow$ any point in $C$;
5      numIncentivized $\leftarrow |\{s_k \in S \mid$ all actions in
         $\mathcal{S}(x^*)$ yield the (weakly) greatest marginal
         payoff for $s_k$ under $\beta'\}|$;
6      if numIncentivized $>$ max then
7          max $\leftarrow$ numIncentivized;
8          $\beta \leftarrow \beta'$;
9      end
10 end
11 return $\beta$;
```

Theorem 4.2. *Assuming a constant number of features, there is a polynomial-time algorithm to find a linear mechanism that incentivizes a maximum number of agents to invest in a specific admissible profile.*

Proof. Using the notation of Kleinberg and Raghavan, for an effort profile x, let $\mathcal{S}(x)$ denote the support of x. Recall that, to incentivize a given profile x^* for all agents, we must ensure all actions in $\mathcal{S}(x^*)$ are weak best responses. Based on our discussion of arrangements above, Algorithm 1 solves this problem in polynomial time when n is constant. Note that the predicate on line 5 is easy to compute for a fixed β', and does not depend on which $\beta' \in C$ is chosen, since whether a given $s_k \in S$ satisfies the predicate is completely determined by the constraints from line 1. □

With very minor adjustments to Algorithm 1, this same technique can be used to solve Problems 6 and 8 as well (see Appendix G).

Theorem 4.3. *Assuming a constant number of features, there is a polynomial-time algorithm to find a linear mechanism that incentivizes a maximum number of agents to choose admissible actions (and consequently, to determine if all agents can be incentivized to choose admissible actions).*

4.2 Hardness Results

Since these algorithms for Problems 6, 7, and 8 all rely on the ability to efficiently enumerate all cells in a low-dimensional hyperplane arrangement, it is natural to ask what happens when the number of features is part of the input, making this technique no longer viable. As it turns out, all three problems are NP-complete in general.

Theorem 4.4. *The following problems are NP-complete:*

1. *Finding a linear mechanism that incentivizes a maximum number of agents to invest only in admissible actions / a specific admissible profile.*
2. *Finding a linear mechanism that incentivizes all agents to invest only in admissible actions.*[9]

Part (1) follows from the same reduction outlined in Section 3.3 since the instances produced by that reduction have a special property: whenever a particular subset of agents can be incentivized to choose admissible actions using a monotone mechanism, they can, in fact, be so incentivized using a linear mechanism. The hardness in part (2) is of a completely different nature, and is proved via a separate reduction from 3SAT. See Appendices D and F for the proofs.

5 Discussion

Designing an evaluation scheme for a group of agents is broad practical dilemma. It comes up in credit scoring, principal-agent relationships without money (commanders and soldiers, teachers and students), employment under collective agreements, etc. In these cases, designing a *single* evaluation rule for all agents is the only realistic approach. This paper addresses the challenge of multi-agent evaluation from a computational perspective, answering an open question of Kleinberg and Raghavan (2019). Our main contribution is in showing that the evaluation problem with more than one agent is “a whole new ball game”: for example, monotone mechanisms now have more power than linear ones, and the goal of incentivizing admissible actions is now separate (and often harder) than incentivizing a particular effort profile.

There are many directions for future research. A natural one is *approximating* the optimal number of incentivized agents when maximizing is NP-hard. Our techniques are able to provide insights in this direction, since we show a close connection to the Feedback Vertex Set problem, for which both approximations and lower bounds are known (Bar-Yehuda et al. 1998). Other future directions include settings with hidden types as well as hidden actions, incentivizing agent cooperation (by allowing features like scores on a group project), or accommodating complex effects of combinations of agent actions.

Acknowledgments

This work was partially supported by the National Science Foundation under grants IIS-1350598, IIS-1714140, CCF-1525932, and CCF-1733556; by the Office of Naval Research under grants N00014-16-1-3075 and N00014-17-1-2428; by a J.P. Morgan AI Research Award; by a Guggenheim Fellowship; by the Israel Science Foundation (grant No. 336/18); and by a Taub Family Foundation Fellowship. Part of this work was done while Dobson was visiting the Technion via the MISTI-MIT Israel Program.

References

Babaioff, M.; Feldman, M.; and Nisan, N. 2006. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, 18–28.

Bar-Yehuda, R.; Geiger, D.; Naor, J.; and Roth, R. M. 1998. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference. *SIAM Journal on Computing* 27(4):942–959.

Dekel, O.; Fischer, F.; and Procaccia, A. D. 2010. Incentive compatible regression learning. *Journal of Computer and System Sciences* 76(8):759–777.

Dütting, P.; Roughgarden, T.; and Talgam-Cohen, I. 2019. Simple versus optimal contracts. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 369–387.

Goodman, J. E., and O’Rourke, J., eds. 1997. *Handbook of Discrete and Computational Geometry*. CRC Press.

Grossman, S. J., and Hart, O. D. 1983. An analysis of the principal-agent problem. *Econometrica* 51(1):7–45.

Hardt, M.; Megiddo, N.; Papadimitriou, C. H.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, 111–122.

Kleinberg, J. M., and Raghavan, M. 2019. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 825–844.

Meir, R.; Procaccia, A. D.; and Rosenschein, J. S. 2008. Strategyproof classification under constant hypotheses: A tale of two functions. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, 126–131.

Meir, R.; Procaccia, A. D.; and Rosenschein, J. S. 2012. Algorithms for strategyproof classification. *Artificial Intelligence* 186:123–156.

Xiao, S.; Wang, Z.; Chen, M.; Tang, P.; and Yang, X. 2020. Optimal common contract with heterogeneous agents. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. To appear.

[9] The hardness of the problem in part (2) of the theorem implies the hardness of the first problem in part (1); we list these separately for consistency with Table 1.