A General Statistical Framework for Designing Strategy-proof Assignment Mechanisms

Harikrishna Narasimhan
Harvard University
Cambridge, MA 02138, USA
hnarasimhan@seas.harvard.edu

David C. Parkes
Harvard University
Cambridge, MA 02138, USA
parkes@eecs.harvard.edu

Abstract

We develop a statistical framework for the design of a strategy-proof assignment mechanism that closely approximates a target outcome rule. The framework can handle settings with and without money, and allows the designer to employ techniques from machine learning to control the space of strategy-proof mechanisms searched over, by providing a rule class with appropriate capacity. We solve a sample-based optimization problem over a space of mechanisms that correspond to *agent-independent price functions* (virtual prices in the case of settings without money), subject to a feasibility constraint on the sample. A transformation is applied to the obtained mechanism to ensure feasibility on all type profiles, and strategy-proofness. We derive a sample complexity bound for our approach in terms of the capacity of the chosen rule class and provide applications for our results.

1 INTRODUCTION

Mechanism design studies situations where a set of self-interested agents each hold private information regarding their preferences over different outcomes. Originating from microeconomic theory, mechanism design has become important in the design of open, algorithmic systems that involve multiple stakeholders. A mechanism receives claims about agent types, selects an outcome, and may additionally charge payments. An important property of a mechanism is that of *strategy-proofness*, where it is in the best interest for each agent to make truthful reports.

The existing theory of mechanism design provides positive and negative results in regard to properties that can be achieved together with strategy-proofness. The theory is quite limited, though, in that:

1) Results are developed for stylized preference domains that may not reflect real-world structure [1].
2) Positive results are limited by an analytical bottleneck that makes analysis difficult in *multi-dimensional type spaces* [2, 3].
3) There are few general methodologies, especially in mechanism design without money, where bespoke mechanisms are developed for a given domain [4].

In practice, one often needs to hand-craft a mechanism based on application-specific requirements. For example,

- **Task assignment:** Consider the problem of designing an assignment mechanism for a ride sharing platform. This is a setting with payments, and the standard mechanism one would use here is the Vickrey-Clarke-Groves (VCG) mechanism. However, if one needs to incorporate specific priority or fairness considerations, the VCG mechanism may not be well-suited, and the designer would be faced with needing to manually design a mechanism that satisfies the requirements.
- **Resource allocation:** Consider the problem of designing a strategy-proof mechanism for the fair allocation of jobs to shared computers based on reported need. This is a setting without money, and a standard strategy-proof mechanism for assignment is random serial dictatorship (RSD). However, if the designer wishes to optimize a different utility criterion than RSD (e.g. a utilitarian objective), then the designer will again have to handcraft a mechanism based on the requirements.

We suppose that alternatively, the designer can provide his requirements in the form of a *target outcome rule* that maps reports to desired outcomes (but is not necessarily strategy-proof). The goal is to automatically design a strategy-proof mechanism that closely mimics this rule, leading to the following question:

Given an arbitrary outcome rule, can we automatically design a strategy-proof mechanism that closely approximates the rule?

The common approach to *automated mechanism design* [5] has been to formulate a search problem over a set of

strategy-proof mechanisms. Some works perform this search over an explicit space of all possible mechanisms, often resulting in intractable optimization problems (with a number of decision variables that grow exponentially in the number of agents). Other approaches search over a parameterized subset of strategy-proof mechanisms [6, 7, 8]. However, these methods are tailored to specific classes of mechanisms known to be strategy-proof. Positive results are available for a Bayesian relaxation of strategy-proofness, and with agent-separable objectives [9, 10], but a more general approach has remained elusive.

In this paper, we develop a general statistical framework for designing strategy-proof mechanisms that closely approximate a target outcome rule. Envisioning settings with abundant data on agent preferences, we assume access to example inputs to a mechanism, each input labeled with a target outcome. Consider for example a setting with an existing, strategy-proof mechanism, with new design requirements specified through target outcomes on historical reports. The goal is to find a strategy-proof mechanism that closely approximates the target outcome rule.

We leverage general, necessary and sufficient conditions for the strategy-proofness of a mechanism, namely an *agent-independence condition*, and a *feasibility requirement* (that the outcome of the mechanism is feasible). Concretely, the framework formulates an optimization problem on a sampled set of agent type profiles over a specified class of outcome rules that satisfy agent-independence. A feasibility constraint is enforced on the sampled profiles, so that the resulting mechanism is feasible on these samples. In addition, we apply a transform to the mechanism to ensure feasibility on all profiles while retaining agent-independence (and thus obtaining strategy-proofness). The particular problem we study is an *assignment problem*, where there is a set of distinct, indivisible items and the outcome assigns at most one item to each agent. The distance to the target assignment is measured in terms of the Hamming distance. The feasibility transform is due to Hashimoto [11] and is well defined for allocation problems.

Unlike previous works on the automated design of strategy-proof mechanisms, our framework neither performs a brute-force search over all mechanisms nor requires the designer to provide a specific parameterized class of strategy-proof mechanisms. Instead we take an intermediate approach, where the space of strategy-proof mechanisms searched over is controlled by the capacity of the outcome class. The capacity of this class can be controlled through standard machine learning techniques, for example by parametrizing the class in a suitable feature space and adjusting the set of features used. By using the general, necessary and sufficient conditions of agent-independence and feasibilty, we remove the need for new characterization results. Rather, the limit of the framework is governed by the limits of the statistical framework.

The main result is an upper bound on the sample complexity of designing strategy-proof mechanisms using our framework. The bound depends on the capacity of the agent-independent function class used, measured in terms of its *Natarajan dimension* D [12]. We show that for n agents and N sampled profiles, the difference in Hamming distance to the target between the designed mechanism and the best strategy-proof mechanism within the hypothesis space is at most:

$$\widetilde{\mathcal{O}}\left(\sqrt{\frac{D}{N}} \; + \; \frac{D}{N}\sum_{i=1}^{n}|\Theta_i|\right),$$

for a distributional assumption, and where $|\Theta_i|$ is the size of the type space for agent i. The linear dependence on $|\Theta_i|$ is a result of the feasibility transformation applied to the sample-optimal rule. This sample complexity is exponentially smaller than the total number of type profiles $\times_{i=1}^{n}|\Theta_i|$.

The proposed approach is quite flexible, in that it can handle settings with and without money. Instantiating the framework to assignment problems, we provide explicit examples of agent-independent rule classes with finite Natarajan dimension that contain feasible assignment rules. For the setting with money, the hypothesis class is defined in terms of agent-independent price functions, with each agent demanding the item that maximizes its utility. For the setting without money, the hypothesis class is defined in terms of virtual price functions and budgets, with each agent demanding its most preferred, affordable item.

1.1 RELATED WORK

The problem of using machine learning to design mechanisms that approximate a target rule was first considered by Procaccia et al. [13] in the context of designing voting rules, but without consideration to strategy-proofness. In the most closely-related work, Dütting et al. [14] use statistical machine learning to design payment rules for a fixed outcome rule. We design both outcome and payment rules, and whereas they provide approximate strategy-proofness, we obtain strategy-proof mechanisms. We also handle mechanism design without money.

Prior work on automated mechanism design adopts specific, parameterized classes of mechanisms [7, 8, 15]. However, these approaches require a designer to have parametric characterizations of strategy-proof mechanisms, and require specialized solvers for each case. More recently, Narasimhan and Parkes [15] consider the problem of using methods from machine learning to design social choice and matching mechanisms that best approximate a target rule, but their approach is also tailored to specific parameterized classes of mechanisms. We provide a more general approach, where the designer only needs to provide a set of rules that satisfy the agent-independence condition.

There has also been previous work that uses statistical or machine learning techniques to design revenue-optimal mechanisms from sampled preference data [16, 17, 18, 19], but this is restricted to settings where the private information of agents is "single-parameter" (roughly, one number, whereas in our setting each agent's type is a value for each item or a rank order on items).

Organization. In Section 2, we begin with the problem setting and in Section 3, describe a general characterization of strategy-proof mechanisms for assignment problems with and without money. In Section 4, we use these characterizations to develop a statistical framework for designing strategy-proof assignment mechanisms. In Section 5, we derive a sample complexity bound for our approach, and in Section 6, we discuss applications of our result to assignment problems with and without money.

2 PROBLEM SETTING

We consider n agents $[n] = \{1, \ldots, n\}$ and m items $[m] = \{1, \ldots, m\}$, and are interested in one-to-one assignments of items to agents. We allow agents to be unassigned, in which case we will say that the agent is assigned to ϕ. An agent may additionally be charged a payment.

We say that an assignment is *feasible* if no two agents are assigned the same item. Let $\Omega \subset [m]^n$ denote the set of feasible one-to-one assignments of items (or ϕ) to the agents. We will use $y \in \Omega$ to denote a feasible assignment and $y_i \in [m]$ for the item allocated to agent i in y.

Each agent is associated with a *type* θ_i from a finite set Θ_i, which is private to the agent. We use $\theta = (\theta_1, \ldots, \theta_n)$ to denote a profile of types, and $\Theta = \times_{i=1}^{n} \Theta_i$ to denote the set of all type profiles. We will use θ_{-i} to denote the profile of types for all but agent i, and $\Theta_{-i} = \times_{j \neq i} \Theta_j$.

In a setting without money, an agent's type induces a preference ordering over items. We will use $o \succ_i o'$ to denote that agent i strictly prefers item $o \in [m]$ over item $o' \in [m]$, and $o \succeq_i o'$ to denote that the agent either strictly prefers o over o' or is indifferent.

In a setting with money, an agent is charged a price for an item, and the agent's type induces a preference ordering over pairs $(o, p_o) \in [m] \times \mathbb{R}_+$ of items and prices. In this case, we will assume quasi-linear preferences. Here each agent i is associated with a valuation function $v_i : \Theta_i \times [m] \to \mathbb{R}_+$, with $v_i(\theta_i, o) \in \mathbb{R}_+$ indicating the value assigned for agent type θ_i to item o. The agent's *utility* for a pair (o, p_o) of items and prices is given by $u_i(\theta_i, (o, p_o)) = v_i(\theta_i, o) - p_o$, and $(o, p_o) \succeq_i (o', p_{o'}) \iff u_i(\theta_i, (o, p_o)) \geqslant u_i(\theta_i, (o', p_{o'}))$.

A *mechanism* receives reports of types from the agents, and maps each agent to an item through an *outcome rule* $f : \Theta \to \Omega$. For a report profile $\hat{\theta}$ from the agents, the assignment to the agents is given by $f(\hat{\theta}) \in \Omega$, and $f_i(\hat{\theta}) \in [m]$ shall denote the item assigned to agent i by f. In settings with money, the mechanism also charges a payment measured in terms of a *payment rule* $p_i : \Theta \to \mathbb{R}_+$.

A desirable property of a mechanism is *strategy-proofness*. A mechanism is strategy-proof if each agent receives its most-preferred outcome (or outcome-price pair) when reporting its true type. More concretely, in a setting without money, a mechanism defined by outcome rule f is strategy-proof if for all $i \in [n]$, $\theta \in \Theta$, and $\theta'_i \in \Theta_i$, $f_i(\theta) \succeq_i f_i(\theta'_i, \theta_{-i})$. Similarly, in a setting with money, a mechanism defined by (f, p) is strategy-proof if for all $i \in [n]$, $\theta \in \Theta$, and $\theta'_i \in \Theta_i$, $(f_i(\theta), p_i(\theta)) \succeq_i (f_i(\theta'_i, \theta_{-i}), p_i(\theta'_i, \theta_{-i}))$. In both cases, let us use $\mathcal{M}_{\text{SP}}$ to denote the space of mechanisms that are strategy-proof.

Agent types are distributed according to an underlying, unknown distribution $\mathcal{D}$ over type profiles Θ. We are provided a *target outcome rule* $g : \Theta \to \Omega$ that need not be strategy-proof, and the goal is to design a strategy-proof mechanism that closely approximates this rule. For this purpose, we adopt a *distance measure* $\ell : [m]^n \times [m]^n \to \mathbb{R}_+$ to measure the distance between the given and target assignments. In particular, we use the normalized *Hamming* distance, $\ell(y, y') = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i \neq y'_i)$ for any $y, y' \in [m]^n$.

The goal is:

$$\min_{(f,p) \in \mathcal{M}'_{\text{SP}}} \mathbf{E}_{\Theta \sim \mathcal{D}} \big[\ell(g(\theta), f(\theta)) \big], \tag{1}$$

where $\mathcal{M}'_{\text{SP}} \subseteq \mathcal{M}_{\text{SP}}$ is some set of strategy-proof mechanisms. In words, we want to find the strategy-proof mechanism in a class of mechanisms that minimizes the expected distance from the target. Since the distribution is over a finite space, an infimum over $\mathcal{M}'_{\text{SP}}$ is always achieved by a mechanism within the class. We shall allow the designer to control the space of strategy-proof mechanisms $\mathcal{M}'_{\text{SP}}$ by providing a suitable rule class with appropriate capacity (or expressive power).

3 CHARACTERIZATION OF STRATEGY-PROOF MECHANISMS

We first provide a general, necessary and sufficient characterization of strategy-proof assignment mechanisms in settings with and without money. These characterizations are standard in mechanism design theory.

Assignment Problem with Money. A mechanism defined by a pair of outcome rule and payment rule (f, p) is strategy-proof iff the following conditions hold [20]:

(1) **Agent independence:** Given the report of the other agents, an agent's prices on each item are independent of its own report. Also, the agent is assigned its most-preferred item, given its report and the agent-independent prices. In other words, the payment

rule $p_i(\theta) = t_i(\theta_{-i}, f_i(\theta))$ for some *price function* $t_i : \Theta_{-i} \times [m] \to \mathbb{R}_+$, and

$$f_i(\theta) \in \operatorname*{argmax}_{o \in [m]} \big\{ v_i(\theta_i, o) - t_i(\theta_{-i}, o) \big\}.$$

(2) **Feasibility:** No two agents get the same item, i.e. for all $i, j \in [n]$,$f_i(\theta) \neq \phi, f_j(\theta) \neq \phi \Rightarrow f_i(\theta) \neq f_j(\theta)$.

In words, an agent cannot change the price for an item by misreporting its type, and given these prices, receives the most-preferred item according to the report. Further, the assignment is feasible for all reports.

Example 1. *The well-known VCG mechanism satisfies the above conditions. In its general form, the VCG mechanism allocates for any report θ an assignment that maximizes welfare (i.e. sum of agent valuations): $f^{vcg}(\theta) \in \operatorname{argmax}_{y \in \Omega} \sum_{i=1}^{n} v_i(\theta_i, y_i)$, and charges each agent i a payment: $p_i^{vcg}(\theta) = H_i(\theta_{-i}) - \sum_{j \neq i} v_j(\theta_j, f_j^{vcg}(\theta))$, where $H_i : \Theta_{-i} \to \mathbb{R}_+$ is a function that is independent of agent i's report. By definition, the VCG mechanism satisfies the feasibility condition. To see that the mechanism also satisfies the agent-independence condition, define $t_i^{vcg} : \Theta_{-i} \times [m] \to \mathbb{R}_+$ for reports θ_{-i} and item o as: $t_i^{vcg}(\theta_{-i}, o) = H_i(\theta_{-i}) - \max_{y \in \Omega, y_i = o} \sum_{j \neq i} v_j(\theta_j, y_j)$. Then it can be verified that $p_i^{vcg}(\theta) = t_i^{vcg}(\theta_{-i}, f_i^{vcg}(\theta))$.*

Assignment Problem without Money. We can obtain a similar characterization by defining a *virtual price function* $t_i^{\text{vir}} : \Theta_{-i} \times [m] \to \mathbb{R}_+$ for each agent. For a given preference profile report $\theta \in \Theta$, we will say that agent i can *afford* item $o \in [m]$ if the virtual price for the item is below a budget of \$1, i.e. $t_i^{\text{vir}}(\theta_{-i}, o) \leqslant 1$. An agent receives one of the items that it can afford. A mechanism defined by outcome rule f is strategy-proof iff the following hold:

(1) **Agent independence:** Given the report of the other agents, an agent's virtual prices are independent of its own report. The agent is assigned its most-preferred item among those it can afford, given its report and the agent-independent prices. In other words, there exists a virtual price function, $t_i^{\text{vir}} : \Theta_{-i} \times [m] \to \mathbb{R}_+$ such that $t_i^{\text{vir}}(\theta_{-i}, f_i(\theta)) \leqslant 1$ and

$$f_i(\theta) \succeq_i o, \forall o \in \{o' \in [m] : t_i^{\text{vir}}(\theta_{-i}, o') \leqslant 1\}.$$

(2) **Feasibility:** No two agents get the same item, i.e. for all $i, j \in [n]$, $f_i(\theta) \neq \phi, f_j(\theta) \neq \phi \Rightarrow f_i(\theta) \neq f_j(\theta)$.

4 A GENERAL STATISTICAL FRAMEWORK

We next introduce a framework that exploits these general, necessary and sufficient characterizations. Specifically, we provide an approach to solve (1) by formulating a sample-based optimization problem over outcome rules that satisfy the above conditions.

We require the designer to provide a class $\mathcal{F}_i$ of functions $f_i : \Theta \to [m]$ that satisfy the *agent independence* condition. For the setting with money, each $f_i \in \mathcal{F}_i$ is required to be of the form $f_i(\theta) \in \operatorname{argmax}_{o \in [m]} \big\{ v_i(\theta_i, o) - t_i(\theta_{-i}, o) \big\}$ for some $t_i : \Theta_{-i} \times [m] \to \mathbb{R}_+$. For the setting without money, each $f_i \in \mathcal{F}_i$ needs to satisfy $t_i^{\text{vir}}(\theta_{-i}, f_i(\theta)) \leqslant 1$ and $f_i(\theta) \succeq_i o, \forall o \in \{o' \in [m] : t_i^{\text{vir}}(\theta_{-i}, o') \leqslant 1\}$ for some $t_i^{\text{vir}} : \Theta_{-i} \times [m] \to \mathbb{R}_+$.

Further, let $\mathcal{F} = \times_{i=1}^{n} \mathcal{F}_i$. We will refer to each function $f_i \in \mathcal{F}_i$ as an agent-independent function, and the concatenated function $f \in \mathcal{F}$ as an agent-independent outcome rule. The outcome rules $f \in \mathcal{F}$ need not satisfy the feasibility condition (i.e. can map a type profile θ to an infeasible assignment $f(\theta) \in [m]^n$), and therefore may not be strategy-proof.

For ease of exposition, we will henceforth assume that *neither the outcome rules $f \in \mathcal{F}$ nor the target rule g leave an agent unassigned* (i.e. do not assign ϕ to an agent). The framework and theoretical results easily extend to the case where this assumption does not hold.

The goal is to solve (1) over all outcome rules in $\mathcal{F}$ that also satisfy the feasibility condition, and find the rule that has minimum Hamming distance from the target rule:

$$\min_{f \in \mathcal{F}} \mathbf{E}_{\theta \sim \mathcal{D}} \big[\ell\big(g(\theta), f(\theta)\big) \big] \tag{2}$$

$$\text{s.t.} \quad f_1(\theta) \neq \ldots \neq f_n(\theta), \ \ \forall \theta \in \Theta.$$

In practice, we do not have access to the type distribution $\mathcal{D}$. Rather, we have a sample $S = \{(\theta^1, y^1 = g(\theta^1)), \ldots, (\theta^N, y^N = g(\theta^N))\} \in (\Theta \times \Omega)^N$ containing agent profiles drawn i.i.d. from $\mathcal{D}$ and labeled according to the target outcome rule g.

We solve an empirical version of the optimization problem (2), with the feasibility constraint enforced only on the profiles in S. A problem is that the obtained rule need not be feasible on type profiles outside S. To address this, we adopt a *feasibility transform* on the obtained rule that ensures that the resulting rule is feasible without compromising agent independence, and thus obtaining strategy-proofness. The two steps of our framework are:

Step I: Constrained Optimization on a Sample. We first solve an empirical version of (2) on sample S:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{k=1}^{N} \ell\big(y^k, f(\theta^k)\big)\big) \tag{3}$$

$$\text{s.t.} \quad f_1(\theta^k) \neq \ldots \neq f_n(\theta^k), \ \ \forall k \in \{1, \ldots, N\}.$$

Step II: Feasibility Transform. The obtained outcome rule need not be feasible on profiles outside S. A naive way to enforce feasibility is to resolve conflicts by canceling an

allocation when there is infeasibility, or through some more sophisticated priority-based approach. But without care, this results in a mechanism that is not strategy-proof; e.g. perhaps an agent can usefully misreport in order to avoid a problem with infeasibility.

We use a transform inspired by an approach due to Hashimoto [11]. For each agent i, we perform the following check: conditioned on the reports of the other agents, *does there exist a type in Θ_i for which the outcome rule would output an infeasible assignment*? If yes, we leave agent i unassigned; otherwise the original assigned item is left unchanged. For the outcome rule $\widehat{f}$ obtained by solving (3), the transform $\mathbb{T}[\widehat{f}] : \Theta \to \Omega$ is given by:

$$\mathbb{T}_i[\widehat{f}](\theta) = \begin{cases} \phi & \text{if } \exists\, \theta_i' \in \Theta_i \text{ s.t. } \widehat{f}(\theta_i', \theta_{-i}) \text{ is infeasible} \\ \widehat{f}_i(\theta) & \text{otherwise,} \end{cases}$$

where $\mathbb{T}_i[\widehat{f}](\theta)$ denotes the item assigned to agent i by the transformed rule. The transform has no effect when $\widehat{f}$ is feasible on all profiles.[1]

Theorem 1 (Hashimoto 2016)**.** *The outcome rule $\mathbb{T}[\widehat{f}]$ is feasible and strategy-proof when $\widehat{f}$ is agent-independent.*

Proof. The transformation $\mathbb{T}$ will leave an agent unassigned whenever its original assignment conflicts with that of the others. Since $\mathbb{T}$ never assigns a new item to an agent, $\mathbb{T}[\widehat{f}]$ is feasible. For strategy-proofness, we consider two cases. (Case 1): the feasibility check for all $\theta_i' \in \Theta_i$ passes, so that agent i's assignment from $\widehat{f}$ is left unperturbed, and no misreport is useful as the agent receives its optimal item given the agent-independent prices. (Case 2): the feasibility check does not pass. But here it would not pass whatever be the report $\hat{\theta}_i$ of agent i, since the test is independent of its report. A misreport is not useful. □

The framework allows a designer to control the space of strategy-proof mechanisms searched over by choosing an appropriately expressive agent-independent rule class $\mathcal{F}$.

Example 2. *We show how the framework can be used to design a strategy-proof mechanism for a simple setting with payments. Consider two homogeneous agents $\{1, 2\}$ and two items $\{1, 2\}$. Assume there are two agent types $\Theta_1 = \Theta_2 = \{\alpha, \beta\}$, with the following valuation functions:*

$$v_1(\alpha, 1) = v_2(\alpha, 1) = 2; \quad v_1(\alpha, 2) = v_2(\alpha, 2) = 1$$
$$v_1(\beta, 1) = v_2(\beta, 1) = 1; \quad v_1(\beta, 2) = v_2(\beta, 2) = 2$$

[1]The transformation does not require an enumeration of all $\times_{i=1}^{n} |\Theta_i|$ type profiles, and performs a check only over the individual type space of a given agent, fixing the reports of the others. It can be implemented with $\sum_{i=1}^{n} |\Theta_i|$ checks.

Suppose the underlying distribution $\mathcal{D}$ over Θ is uniform, and the target rule the designer wants to approximate is:

$$g(\alpha, \alpha) = (1, 2); \qquad g(\beta, \alpha) = (1, 2)$$
$$g(\alpha, \beta) = (1, 2); \qquad g(\beta, \beta) = (2, 1)$$

Assume the designer provides a class $\mathcal{F}_i$ of agent-independent functions $f_i(\theta) = \text{argmax}_{o \in [m]}\{v_i(\theta_i, o) - t_i(\theta_{-i}, o)\}$, with the following two candidates for the payment function $t_i : \Theta_{-i} \times \{1, 2\} \to \mathbb{R}_+$:

τ^A:

	1	2
α	1	0
β	0	2

τ^B:

	1	2
α	1	0
β	0	1

Assume we are provided a training sample with two randomly drawn type profiles and labeled with the target outcomes: $S = \{((\alpha, \alpha), (1, 2)), ((\alpha, \beta), (1, 2))\}$. We now go over the two steps of the framework:

I: Constrained optimization over sample. *We first solve the optimization problem in* (3) *over the given hypothesis class. Note that an outcome rule $\widehat{f}^A$ constructed using $t_1 = t_2 = \tau^A$ (with ties broken in favor of the smaller item for agent 1, and larger item for agent 2) is a solution to* (3) *as it is both feasible on S and yields zero error. On the first type profile (α, α), this rule gives us*

$$\widehat{f}_1^A(\alpha, \alpha) = \text{argmax}_{o \in \{1,2\}}\{v_1(\alpha, o) - \tau^A(\alpha, o)\} = 1$$
$$\widehat{f}_2^A(\alpha, \alpha) = \text{argmax}_{o \in \{1,2\}}\{v_2(\alpha, o) - \tau^A(\alpha, o)\} = 2$$

and on the second type profile, we get

$$\widehat{f}_1^A(\alpha, \beta) = \text{argmax}_{o \in \{1,2\}}\{v_1(\alpha, o) - \tau^A(\beta, o)\} = 1$$
$$\widehat{f}_2^A(\alpha, \beta) = \text{argmax}_{o \in \{1,2\}}\{v_2(\beta, o) - \tau^A(\alpha, o)\} = 2$$

II: Feasibility transformation. *The outcome rule is not necessarily feasible on a type profile outside S. For example, on the type profile (β, β), the rule outputs an infeasible allocation $(1, 1)$. As a second step, we apply the feasibility transform to enforce feasibility without violating the agent-independence property. The resulting outcome rule $\mathbb{T}[\widehat{f}^A]$ can then be verified to yield the following:*

$$\mathbb{T}[\widehat{f}^A](\alpha, \alpha) = (1, 2); \quad \mathbb{T}[\widehat{f}^A](\alpha, \beta) = (1, \phi)$$
$$\mathbb{T}[\widehat{f}^A](\beta, \alpha) = (\phi, 1); \quad \mathbb{T}[\widehat{f}^A](\beta, \beta) = (\phi, \phi)$$

On the other hand, if we were provided a larger sample, say $S' = \{((\beta, \alpha), (1, 2)), ((\alpha, \beta), (1, 2)), ((\beta, \beta), (2, 1)), ((\alpha, \beta), (1, 2))\}$, then the outcome rule $\widehat{f}^A$ is no longer feasible on the sample. In this case, we would instead pick $t_1 = t_2 = \tau^B$. It can be verified that the resulting rule $\widehat{f}^B$ is the VCG outcome rule. This rule is feasible on all profiles, and the transform has no effect: $\mathbb{T}[\widehat{f}^B] = \widehat{f}^B$.

5 SAMPLE COMPLEXITY GUARANTEE

A potential concern is the extent to which the feasibility transform reduces the quality of the solution obtained by solving (3). We shall see that under an assumption on distribution $\mathcal{D}$, and with a sufficiently large sample, the transformed rule becomes arbitrarily close to the best strategy-proof approximation in $\mathcal{F}$. This holds when each $\mathcal{F}_i$ has finite capacity, as we elaborate in this section.

To have any hope of solving our original optimization problem in (1) using a finite sample, we will require the space of agent-independent functions $\mathcal{F}_i$ to have limited capacity, so that the obtained outcome rule does not overfit the sample. The specific notion of capacity we consider is the *Natarajan dimension*, commonly used while analyzing generalization performance of multi-class classifiers [12] (note that each $f_i \in \mathcal{F}_i$ can be seen as a multiclass classifier mapping type profiles to one of m items).

Definition 1 (**Natarajan dimension**)**.** A set of profiles $A \subseteq \Theta$ is said to be N-shattered by $\mathcal{F}_i$ if there exists labelings $L_1, L_2 : A \to [m]$ such that $L_1(\theta) \neq L_2(\theta),\ \forall \theta \in A$, and for every subset $B \subseteq A$, there is a $f_i \in \mathcal{F}_i$ such that $f_i(\theta) = L_1(\theta),\ \forall \theta \in B$ and $f_i(\theta) = L_2(\theta),\ \forall \theta \in A \backslash B$. The Natarajan dimension of $\mathcal{F}_i$ is the size of the largest set A that is N-shattered by $\mathcal{F}_i$.

The Natarajan dimension is analogous to the *VC dimension* used in binary classification settings. In fact, for binary hypothesis classes, this quantity is the same as the VC dimension (with L_1 and L_2 being the all 1's and all 0's labelings respectively). As with the VC dimension, finiteness of the Natarajan dimension is necessary and sufficient for learnability of a multiclass hypothesis class (see for example [21]). Hence, we assume that each agent-independent class $\mathcal{F}_i$ has finite Natarajan dimension. We will further require a smoothness assumption on distribution $\mathcal{D}$:

Assumption A. Let μ be the p.m.f. associated with distribution $\mathcal{D}$. There exists $\alpha \geqslant 1$ such that for all i, $\theta_i', \theta_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}, \mu(\theta_i', \theta_{-i}) \leqslant \alpha \mu(\theta_i, \theta_{-i})$.

Assumption A requires that type profiles that differ only in one coordinate have similar probability masses. The value of α measures the closeness of the type distribution to a uniform distribution (with higher values indicating the distribution is father away from being uniform). For the uniform distribution we have $\alpha = 1$. Assumption A is used to enable an analysis of the effect of the feasibility transform on the outcome rule.

Each outcome rule in $\mathcal{F}$ satisfies the agent-independence condition. Let $\mathcal{F}_{\text{SP}} \subseteq \mathcal{F}$ be the subset of rules that also satisfy the feasibility condition, and are thus strategy-proof, i.e. outcome rules in $f \in \mathcal{F}$ that are feasible on all $\theta \in \Theta$. We only consider function classes that contain at least one feasible rule, i.e. for which $\mathcal{F}_{\text{SP}} \neq \phi$. Our goal is to find a rule in $\mathcal{F}_{\text{SP}}$ that best approximates target rule g.

Our approach picks a rule $\widehat{f}$ that is feasible on sample S, but need not be feasible on type profiles outside S. The transformation $\mathbb{T}$ ensures feasibility on all profiles, while ensuring strategy-proofness. We show that the transformed rule $\mathbb{T}[\widehat{f}]$ converges in the large sample limit to the best rule in $\mathcal{F}_{\text{SP}}$:

Theorem 2. *Let $\mathcal{D}$ satisfy Assumption A, and assume $\mathcal{F}_{\text{SP}} \neq \varnothing$. Let $\widehat{f}$ denote the rule obtained by solving* (3) *on a sample S of size N, and $\tilde{f} = \mathbb{T}[\widehat{f}]$. Then with probability at least $1 - \delta$ (over draw of S from $\mathcal{D}^N$),*

$$\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \tilde{f}(\theta)\big)\big] \leqslant \min_{f\in\mathcal{F}_{\text{SP}}} \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big] + \widetilde{\mathcal{O}}\left(\sqrt{\frac{D}{N}}\right) + \widetilde{\mathcal{O}}\left(\frac{\alpha D}{N}\sum_{i=1}^{n}|\Theta_i|\right),$$

where D is an upper bound on the Natarajan dimension of each agent-independent function class $\mathcal{F}_i$, and $\widetilde{O}$ hides terms that are logarithmic in n, m, N, D and δ.

The first term arises from $\widehat{f}$ yielding minimum error on sample S, and decreases with increasing sample size N.

The second term captures the effect of the feasibility transformation $\mathbb{T}$, and has a linear dependence on the size of an agent's type space $|\Theta_i|$, while being exponentially smaller than the total number of type profiles $\times_{i=1}^{n}|\Theta_i|$. This term also decreases with sample size N; this is because as N increases, $\widehat{f}$ becomes feasible on a larger fraction of the population, the effect of the transformation $\mathbb{T}$ becomes smaller. Thus for a finite class capacity D, both the above terms go to 0 as $N \to \infty$, and the transformed outcome rule $\widetilde{f}$ converges to the optimal rule in $\mathcal{F}_{\text{SP}}$.

The larger the class capacity D, the larger is the space of strategy-proof rules searched over. However, as seen in the above bound, this will also lead to a larger bias due to overfitting. Thus based on the size of the available sample, the designer needs to appropriately tune the capacity of the agent-independent class, so as to strike a trade-off between the size of the strategy-proof hypothesis space searched over, and the corresponding bias introduced.

5.1 PROOF

We give the proof for Theorem 2. Let $\mathcal{F}_S$ denote the subset of all agent-independent rules in $\mathcal{F}$ that are feasible on S. Recall that the outcome rule $\widehat{f} \in \mathcal{F}$ obtained by solving (3) is in $\mathcal{F}_S$, and also yields the minimum sample error over all rules in $\mathcal{F}_S$. Note that $\mathcal{F}_{\text{SP}}$ is a subset of the rules $\mathcal{F}_S$ that are feasible on all type profiles, and thus strategy-proof:

$$\boxed{\mathcal{F}_{\text{SP}} \subseteq \mathcal{F}_S \subseteq \mathcal{F}}$$

Also, note that the final transformed outcome rule $\widetilde{f} = \mathbb{T}[\widehat{f}]$ is feasible on all profiles, but may not be a rule in $\mathcal{F}_{\text{SP}}$.

We show that with increasing sample size, $\widetilde{f}$ converges to the rule in $\mathcal{F}_{\text{SP}}$ that best approximates the target. While we assume neither the rules $f \in \mathcal{F}$ nor the target rule g assign ϕ to an agent, the transformation $\mathbb{T}$ is allowed to cancel an item to an agent. Hence, while evaluating the designed mechanism against the target outcome rule, an assignment of ϕ to an agent will be counted as an error.

The proof is based on uniform convergence arguments commonly used in the generalization analysis of multiclass classifiers. We will make use of the fact that $\widehat{f}$ is chosen from a finite capacity rule class. We first show that since $\widehat{f}$ minimizes the sample error over all rules in $\mathcal{F}_S$, its expected error to the target is also close to the least possible error within $\mathcal{F}_S$, and in turn to the least error within the subset of feasible and strategy-proof rules $\mathcal{F}_{\text{SP}} \subseteq \mathcal{F}_S$. However, the final transformed rule $\tilde{f}$ may be different from $\widehat{f}$, as it can cancel items assigned by $\widehat{f}$. We further show that since $\widehat{f}$ is feasible on sample S, it is also feasible on a large portion of the population; this together with our smoothness assumption on the distribution implies that the expected error of $\tilde{f}$ is close to that of $\widehat{f}$. Thus we are able to bound the expected error of $\tilde{f}$ in terms of the minimum error within $\mathcal{F}_{\text{SP}}$, and a sample complexity term.

In particular, we analyze three quantities:

$$\begin{aligned}
\epsilon_{\text{err}} &= \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \widehat{f}(\theta)\big)\big], \\
\epsilon_{\text{infeasible}} &= \mathbf{P}_{\theta\sim\mathcal{D}}\big(\widehat{f}_i(\theta) = \widehat{f}_j(\theta) \text{ for some } i, j \in [n]\big), \\
\epsilon^i_{\mathbb{T}} &= \mathbf{P}_{\theta\sim\mathcal{D}}\big(\widetilde{f}_i(\theta) = \phi\big).
\end{aligned}$$

Here ϵ_{err} is the expected distance of the untransformed rule $\widehat{f}$ from the target; $\epsilon_{\text{infeasible}}$ is the probability of $\widehat{f}$ being infeasible on a random profile; and $\epsilon^i_{\mathbb{T}}$ is the probability that the transformation $\mathbb{T}$ cancels the item assigned to agent i.

Since the transformation $\mathbb{T}$ leaves $\widehat{f}_i$ unchanged on all but a fraction $\epsilon^i_{\mathbb{T}}$ of the profiles, the distance of the final transformed outcome rule $\widetilde{f}$ from the target can be bounded in terms of the error of the untransformed rule ϵ_{err}, and $\epsilon^i_{\mathbb{T}}$'s:

$$\begin{aligned}
&\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \tilde{f}(\theta)\big)\big] \;=\; \mathbf{E}_{\theta\sim\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\big(\widetilde{f}_i(\theta) \neq g_i(\theta)\big)\right] \\
&= \mathbf{E}_{\theta\sim\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\big(\widetilde{f}_i(\theta) = \widehat{f}_i(\theta) \neq g_i(\theta)\big)\right] \\
&\qquad + \mathbf{E}_{\theta\sim\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\big(\widetilde{f}_i(\theta) = \phi \neq g_i(\theta)\big)\right] \\
&\leqslant \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \widehat{f}(\theta)\big)\big] \;+\; \mathbf{E}_{\theta\sim\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\big(\widetilde{f}_i(\theta) = \phi\big)\right] \\
&= \epsilon_{\text{err}} \;+\; \frac{1}{n}\sum_{i=1}^{n}\epsilon^i_{\mathbb{T}}. \qquad (4)
\end{aligned}$$

We bound ϵ_{err} and $\epsilon^i_{\mathbb{T}}$. For this, we will in turn require a bound on $\epsilon_{\text{infeasible}}$. We start with an outline:

- *Bounding ϵ_{err} (Lemma 3):* We show that the *expected* distance of this rule from the target is close to that of the optimal rule in $\mathcal{F}_{\text{SP}}$.
- *Bounding $\epsilon_{\text{infeasible}}$ (Lemma 4):* We show that $\widehat{f}$ is feasible on a large fraction of the population.
- *Bounding $\epsilon^i_{\mathbb{T}}$ (Lemma 5):* We use the smoothness assumption on $\mathcal{D}$ (Assumption A) to show that the transformation $\mathbb{T}$ will have limited effect on $\widehat{f}$ as long as $\widehat{f}$ is feasible on a large portion of the population. In particular, we bound each $\epsilon^i_{\mathbb{T}}$ in terms of $\epsilon_{\text{infeasible}}$.

We begin by bounding ϵ_{err}. It is useful to state a generalization bound on the difference between the empirical and population errors of an outcome rule f chosen from a class of finite Natarajan dimension [21] (see Lemma 10 in Appendix A). W.p. $\geqslant 1-\delta$ (over draw of S), $\forall f \in \mathcal{F}$,

$$\begin{aligned}
&\left|\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big] - \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, f(\theta^k)\big)\right| \\
&\qquad\qquad \leqslant \;\; \mathcal{O}\left(\sqrt{\frac{D\ln(m) + \ln(n/\delta)}{N}}\right). \qquad (5)
\end{aligned}$$

We then have:

Lemma 3. *Fix $\delta > 0$. With probability at least $1-\delta$ (over draw of S from $\mathcal{D}^N$),*

$$\begin{aligned}
\epsilon_{\text{err}} \;\leqslant\; &\min_{f\in\mathcal{F}_{\text{SP}}}\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big] \\
&\qquad + \mathcal{O}\left(\sqrt{\frac{D\ln(m) + \ln(n/\delta)}{N}}\right).
\end{aligned}$$

Proof. Denote $f^* \in \operatorname{argmin}_{f\in\mathcal{F}_{\text{SP}}}\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big]$. We wish to bound:

$$\begin{aligned}
&\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \widehat{f}(\theta)\big)\big] - \min_{f\in\mathcal{F}_{\text{SP}}}\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big] \\
&= \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \widehat{f}(\theta)\big)\big] \;-\; \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, \widehat{f}(\theta^k)\big) \\
&\qquad + \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, \widehat{f}(\theta^k)\big) \;-\; \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f^*(\theta)\big)\big] \\
&\leqslant \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), \widehat{f}(\theta)\big)\big] \;-\; \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, \widehat{f}(\theta^k)\big) \\
&\qquad + \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, f^*(\theta^k)\big) \;-\; \mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f^*(\theta)\big)\big] \\
&\leqslant 2\sup_{f\in\mathcal{F}}\left|\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta), f(\theta)\big)\big] \;-\; \frac{1}{N}\sum_{k=1}^{N}\ell\big(y^k, f(\theta^k)\big)\right|,
\end{aligned}$$

where the second step uses the fact that $\widehat{f}$ has minimum empirical error on S over all $\mathcal{F}_S \supseteq \mathcal{F}_{\text{SP}}$ and hence a lesser or equal empirical error compared to $f^* \in \mathcal{F}_{\text{SP}}$; the last step uses the fact that both $\widehat{f}, f^* \in \mathcal{F}$. The generalization bound in (5) then gives the desired result. □

We next focus on bounding $\epsilon_{\text{infeasible}}$.

Lemma 4. *Assume $\mathcal{F}_{\text{SP}} \neq \varnothing$. Fix $\delta > 0$. Then w.p. at least $1-\delta$ (over draw of S from $\mathcal{D}^N$),*

$$\epsilon_{\text{infeasible}} \leqslant \mathcal{O}\left(\frac{nD\ln(mnD)\ln(N)+\ln(1/\delta)}{N}\right).$$

Proof. (Sketch) We provide the full proof in Appendix B.1. For any $f : \Theta \to \Omega$, define a binary function $G_f : \Theta \to \{0,1\}$ as $G_f(\theta) = \mathbf{1}\big(f_1(\theta) \neq \ldots \neq f_n(\theta)\big)$. Clearly, f is feasible iff G_f evaluates to 1 on all type profiles. Since $\mathcal{F}_{\text{SP}} \neq \phi$, there always exists a f in $\mathcal{F}$ which is feasible, and hence there always exists a G_f which outputs 1 on all profiles. Treating G_f as a binary classifier, one can now appeal to standard VC dimension based learnability results for classification [22], with the loss function being the 0-1 loss against the all 1's labeling. The VC dimension of the class of all functions $\{G_f : \Theta \to \{0,1\} : f \in \mathcal{F}\}$ can be shown to be at most $\mathcal{O}(nD\ln(nmD))$. Then w.p. $\geqslant 1-\delta$,

$$\begin{aligned}\epsilon_{\text{infeasible}} &= \mathbf{E}_{\theta\sim\mathcal{D}}\big[\mathbf{1}\big(G_{\hat{f}}(\theta) \neq 1\big)\big] \\ &\leqslant \mathcal{O}\left(\frac{nD\ln(nmD)\ln(N)+\ln(1/\delta)}{N}\right),\end{aligned}$$

which implies the statement of the lemma. □

We finally bound $\epsilon^i_{\mathbb{T}}$ in terms $\epsilon_{\text{infeasible}}$.

Lemma 5. *Under Assumption A, $\epsilon^i_{\mathbb{T}} \leqslant \alpha|\Theta_i|\epsilon_{\text{infeasible}}$.*

Proof. Let us use μ to denote the p.m.f. associated with distribution $\mathcal{D}$. Also, let $\Theta_{\text{infeasible}} \subseteq \Theta$ denote the subset of type profiles on which $\widehat{f}$ is infeasible, i.e. type profiles $\theta \in \Theta$ for which $\widehat{f}_i(\theta) = \widehat{f}_j(\theta)$ for some i and j. Clearly, $\epsilon_{\text{infeasible}} = \sum_{\theta\in\Theta_{\text{infeasible}}} \mu(\theta)$. Further, note that the set of type profiles on which the transformation $\mathbb{T}$ makes a null allocation to agent i is precisely the set of type profiles that are one hop away (i.e. differ in agent i's type) from those in $\Theta_{\text{infeasible}}$. Therefore,

$$\begin{aligned}\epsilon^i_{\mathbb{T}} &= \sum_{\theta\in\Theta}\mu(\theta)\mathbf{1}\big(\tilde{f}_i(\theta)=\phi\big) = \sum_{\theta\in\Theta_{\text{infeasible}}}\sum_{\theta'_i\in\Theta_i}\mu(\theta'_i,\theta_{-i}) \\ &\leqslant \sum_{\theta\in\Theta_{\text{infeasible}}}\sum_{\theta'_i\in\Theta_i}\alpha\mu(\theta) = \alpha\sum_{\theta'_i\in\Theta_i}\sum_{\theta\in\Theta_{\text{infeasible}}}\mu(\theta) \\ &= \alpha|\Theta_i|\epsilon_{\text{infeasible}},\end{aligned}$$

where the inequality follows from Assumption A. □

Combining Lemmas 4-5 with (4) gives us w.p. at least $1-\delta$ (over draw of S):

$$\begin{aligned}\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta),\tilde{f}(\theta)\big)\big] &\leqslant \min_{f\in\mathcal{F}_{\text{SP}}}\mathbf{E}_{\theta\sim\mathcal{D}}\big[\ell\big(g(\theta),f(\theta)\big)\big] \\ &+\mathcal{O}\left(\sqrt{\frac{D\ln(m)+\ln(n/\delta)}{N}}\right) \\ &+\mathcal{O}\left(\frac{\alpha}{n}\sum_{i=1}^{n}|\Theta_i|\frac{nD\ln(nmD)\ln(N)+\ln(1/\delta)}{N}\right).\end{aligned}$$

This completes the proof of Theorem 2.

6 APPLICATIONS

We provide instantiations of the framework to assignment problems with and without money. In each case, we construct examples of agent-independent function classes $\mathcal{F}_i$ that have finite Natarajan dimension. We also show that these function classes can be used to model feasible outcome rules, which together with the agent-independence property are guaranteed to be strategy-proof.

6.1 ASSIGNMENT PROBLEM WITH MONEY

As noted in Section 3, a strategy-proof outcome rule in this setting is necessarily of the following form for an agent-independent price rule $t_i : \Theta_{-i} \to \mathbb{R}_+$.

$$f_i(\theta) \in \operatorname*{argmax}_{o\in[m]}\big\{v_i(\theta_i,o) - t_i(\theta_{-i},o)\big\}.$$

One way to construct an agent-independent function class for this setting is by modeling the above price rule t_i as a linear function in a suitable feature space, i.e. as $t^{\mathbf{w}}_i(\theta_{-i},o) = \mathbf{w}_i^\top\Psi_i(\theta_{-i},o)$ for some model vector $\mathbf{w}_i \in \mathbb{R}^d$ and feature map $\Psi_i : \Theta_{-i}\times[m] \to \mathbb{R}^d$. Let $\bar{\mathcal{F}}^{\Psi}_i$ be the corresponding class of agent-independent functions obtained for different model vectors $\mathbf{w}_i$. This class resembles the class of linear discriminant classifiers, which is known to have a finite Natarajan dimension [21]:

Theorem 6. *The Natarajan dimension of $\bar{\mathcal{F}}^{\Psi}_i$ is at most $\mathcal{O}(d\ln(d))$.*

Note that one can fine-tune the capacity of this class by adjusting the number of features d used. Below, we show that the function class admits feasible and strategy-proof outcome rules for an appropriate choice of feature map.

Example feature map. We describe a feature map with two parts, inspired by the VCG price function seen in Example 1. For an agent i and item o, the first part contains the valuations for all agents other than i:

$$\Psi^1_i(\theta_{-i},o) = [v_j(\theta_j,o')]^m_{o'=1}]_{j\neq i} \in \mathbb{R}_+^{(n-1)\times m}.$$

The second part of the feature map contains the valuations for the other agents when they receive the *welfare-maximizing* assignment from items other than o:

$$\Psi^2_i(\theta_{-i},o) = [v_j(\theta_j,y_j^{\setminus i,o})]_{j\neq i} \in \mathbb{R}_+^{n-1},$$

where $y^{\setminus i,o} \in \operatorname{argmax}_{y\in\Omega,\, y_i=o}\sum_{j\neq i}v_j(\theta_j,y_j)$.

The feature map $\bar{\Psi}_i(\theta_{-i},o) = [\Psi^1_i(\theta_{-i},o),\Psi^2_i(\theta_{-i},o)]$ then allows us to construct feasible outcome rules.

Theorem 7. *There exists $\mathbf{w}_i \in \mathbb{R}^{(n-1)(m+1)}$ s.t. the prices $t^{\mathbf{w}}_i(\theta_{-i},o) = \mathbf{w}_i^\top\bar{\Psi}_i(\theta_{-i},o)$ are non-negative and yield a feasible outcome rule for a suitable tie-breaking scheme.*

The specific model vector we consider is

$$\mathbf{w}_i = [\underbrace{1, 1, \ldots, 1}_{(n-1)\times m}, \underbrace{-1, -1, \ldots, -1}_{n-1}].$$

The first part of the model vector ensures that the payments are non-negative, while the second part leads to a *VCG-style* welfare maximizing outcome rule. The proof details are provided in Appendix B.2.

6.2 ASSIGNMENT PROBLEM WITHOUT MONEY

In this setting, a strategy-proof outcome rule satisfied the following for some agent-independent virtual price functions $t_i^{\text{vir}} : \Theta_{-i} \times [m] \to \mathbb{R}_+$: $\quad t_i^{\text{vir}}(\theta_{-i}, f_i(\theta)) \leqslant 1$ and

$$f_i(\theta) \succeq_i o,\ \forall o \in \{o' \in [m] \,:\, t_i^{\text{vir}}(\theta_{-i}, o') \leqslant 1\}.$$

As before, to construct an agent-independent function class, we can model the virtual price function t_i^{vir} as a linear function $t_i^{\text{vir},\mathbf{w}}(\theta_{-i}, o) = \mathbf{w}_i^\top \Psi_i(\theta_{-i}, o)$ for some model vector $\mathbf{w}_i \in \mathbb{R}^d$ and feature map $\Psi_i : \Theta_{-i} \times [m] \to \mathbb{R}^d$. Let $\widetilde{\mathcal{F}}_i^{\Psi}$ be the corresponding class of agent-independent functions obtained for different model vectors $\mathbf{w}_i$.

Unlike the setting with money, here the payment functions do not directly resemble standard classification constructs. Below, we derive a bound on the Natarajan dimension of the proposed function class (see Appendix B.3 for proof).

Theorem 8. *The Natarajan dimension of $\widetilde{\mathcal{F}}_i^{\Psi}$ is at most $\mathcal{O}((md)\ln(md))$.*

Thus the capacity of the function class is finite and can be tuned by varying the number of features used. Also, this class includes feasible and strategy-proof outcome rules for an appropriate choice of the feature map.

Example feature map. Define for agent i, a function $\text{rank}_i : \Theta_i \times [m] \to [m]$ that maps a type θ_i and item o to the number of items that the agent prefers less than o: $\text{rank}(\theta_i, o) = \sum_{o'=1}^{m} \mathbf{1}(o \succ_i o')$ (note higher ranks imply greater preference to item o). For an agent i and item o, the prescribed feature map is then a $n \times m$ binary encoding of the ranks assigned by agents other than i to item o:

$$\widetilde{\Psi}_i(\theta_{-i}, o)[j, k] = \begin{cases} \mathbf{1}(\text{rank}_j(\theta_j, o) = k) & \text{if } j \neq i \\ 0 & \text{otherwise} \end{cases},$$

where we use $[j, k]$ to denote the index $(j-1)m + k$.

Theorem 9. *There exists $\mathbf{w}_i \in \mathbb{R}^{n\times m}$ such that the virtual price functions $t_i^{vir,\mathbf{w}}(\theta_{-i}, o) = \mathbf{w}_i^\top \widetilde{\Psi}_i(\theta_{-i}, o)$ yield a feasible outcome rule.*

In particular, a *serial dictatorship (SD) style* feasible, and strategy-proof outcome rule is within this class. Fix a priority $\pi : [n] \to [n]$ over the agents, where $\pi(i)$ denotes the priority to agent i (with 1 indicating the lowest priority, and n indicating the highest). The following vector $\mathbf{w}_i \in \mathbb{R}^{n\times m}$ then yields a SD style rule with priority ordering π: for any $j \in [n], k \in [m]$,

$$w_i[j, k] = \begin{cases} 2 & \pi(j) > \pi(i),\ k \geqslant m - n + \pi(j) \\ 0 & \text{otherwise} \end{cases}.$$

The proof of feasibility is provided in Appendix B.4.

7 CONCLUSION AND OPEN QUESTIONS

We have developed a general statistical framework for designing strategy-proof mechanisms that closely approximate a given target outcome rule. Our approach does not require domain-specific characterizations, and only requires the designer to provide a class of rules that satisfy a simple agent-independent condition. By tuning the capacity of this class, one can control the space of strategy-proof mechanisms optimized over. We have provided sample complexity bounds for our method and instantiated applications to assignment problems with and without money.

There are several questions that arise from our work:

- The optimization problem (3) can be formulated and solved as a mixed integer linear program. But, how can one solve this problem efficiently in practice? For the setting with money, the problem can be solved approximately by adopting *convex relaxations* from machine learning (see [14]). It will be interesting to understand the effectiveness of these relaxations as well as to identify similar relaxations for the setting without money.

- How can the framework be extended to *infinite type spaces*, and how can the feasibility transformation be implemented efficiently in such a setting?

- How can the framework be extended to *instance-dependent* distance functions, and applied to specific design objectives such as welfare or revenue?

- The sample complexity result requires that the agent-independent function classes have finite capacity. Can we use an approach similar to structural risk minimization to incorporate a *universal hypothesis class* in the framework, and thus cover the entire space of strategy-proof mechanisms?

Acknowledgments. The authors acknowledge the anonymous reviewers for their comments. The authors thank Shivani Agarwal for helpful discussions. HN thanks Harish G. Ramaswamy for a helpful discussion.

References

[1] M.O. Jackson. Mechanism theory. In U. Derigs, editor, *Optimization and Operations Research*. EOLSS Publishers, 2003.

[2] J.C. Rochet. A necessary and sufficient condition for rationalizability in a quasilinear context. *J. Math. Econ.*, 16:191–200, 1987.

[3] R. Lavi and C. Swamy. Truthful mechanism design for multidimensional scheduling via cyclic monotonicity. *Games and Economic Behavior*, 67:99–124, 2009.

[4] J. Schummer and R.V. Vohra. Mechanism design without money. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10. Cambridge University Press, 2007.

[5] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *UAI*, 2002.

[6] V. Conitzer and T. Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *AAMAS*, 2004.

[7] M. Guo and V. Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *AAAI*, 2010.

[8] X. Sui, C. Boutilier, and T. Sandholm. Analysis and optimization of multi-dimensional percentile mechanisms. In *IJCAI*, 2013.

[9] Y. Cai, C. Daskalakis, and S.M. Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *FOCS*, 2012.

[10] S. Alaei, H. Fu, N. Haghpanah, J.D. Hartline, and A. Malekian. Bayesian optimal auctions via multi- to single-agent reduction. In *EC*, 2012.

[11] T. Hashimoto. The generalized random priority mechanism with budgets. Technical report, Yeshiva University, 2016.

[12] B. K. Natarajan. On learning sets and functions. *Maching Learning*, 4(1), 1989.

[13] A.D. Procaccia, A. Zohar, Y. Peleg, and J.S. Rosenschein. The learnability of voting rules. *Artificial Intelligence*, 173:1133–1149, 2009.

[14] P. Dütting, F.A. Fischer, P. Jirapinyo, J.K. Lai, B. Lubin, and D.C. Parkes. Payment rules through discriminant-based classifiers. *ACM Trans. Economics and Comput.*, 3(1):5, 2015.

[15] H. Narasimhan and D.C. Parkes. Automated mechanism design without money via machine learning. In *IJCAI*, 2016.

[16] M.-F. Balcan, A. Blum, J.D. Hartline, and Y. Mansour. Mechanism design via machine learning. In *FOCS*, 2005.

[17] M. Mohri and A.M. Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *ICML*, 2014.

[18] R. Cole and T. Roughgarden. The sample complexity of revenue maximization. In *STOC*, pages 243–252, 2014.

[19] J.H. Morgenstern and T. Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *NIPS*, 2015.

[20] N. Nisan. Introduction to Mechanism Design. In N. Nisan, T. Roughgarden, É. Tardos, and V.V. Vazirani, editors, *Algorithmic Game Theory*, pages 209–241. Cambridge University Press, 2007.

[21] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the erm principle. In *COLT*, 2011.

[22] M. Anthony and P.L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999.