Machine learning for dynamic incentive problems*

Philipp Renner
Department of Economics
University of Lancaster
p.renner@lancaster.ac.uk

Simon Scheidegger
Department of Banking and Finance
University of Zurich
simon.scheidegger@uzh.ch

This version: January 17, 2018
Current version: this link

Abstract

We propose a generic method for solving infinite-horizon, discrete-time dynamic incentive problems with hidden states. We first combine set-valued dynamic programming techniques with Bayesian Gaussian mixture models to determine irregularly shaped equilibrium value correspondences. Second, we generate training data from those pre-computed feasible sets to recursively solve the dynamic incentive problem by a massively parallelized Gaussian process machine learning algorithm. This combination enables us to analyze models of a complexity that was previously considered to be intractable. To demonstrate the broad applicability of our framework, we compute solutions for models of repeated agency with history dependence, many types, and varying preferences.

Keywords: Dynamic Contracts, Principal-Agent Model, Dynamic Programming, Machine Learning, Gaussian Processes, High-Performance Computing.

JEL Classification: C61, C73, D82, D86, E61

*We thank Johannes Brumm, Ken Judd, Felix Kubler, Luca Mazzone, John Rust, Karl Schmedders, Sevin Yeltekin, and seminar participants at the University of Zurich for their valuable comments. This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s790. Simon Scheidegger gratefully acknowledges support from PASC.

1 Introduction

Dynamic incentive (DI) problems are of key importance in model-based economics. They occur whenever two parties form a contract with asymmetric information, and encompass questions such as manager remuneration, insurance contracts, and optimal taxation (see, e.g., Golosov et al. (2016) and references therein for a thorough review). Solving these models is a formidable task, as standard recursive techniques (see, e.g., Stokey et al. (1989)) are often not directly applicable. To this end, previous research both into discrete-time[1] and into continuous-time settings[2] extensively studied methods, in various contexts, of recursively representing incentive-compatible contracts in order to make them formally tractable. There are two types of obstacle that arise in these models in general. First, unobservable continuous actions lead to incentive constraints that are are themselves optimization problems. In the literature, they are commonly referred to as *moral hazard* problems. Second, in so-called *adverse selection* problems, unobserved discrete states make it necessary to look at a high-dimensional dynamic program with an unknown domain.

In this paper, we present a novel, generic method for dealing with DI problems. In particular, we focus on solving infinite-horizon, discrete-time DI models with hidden states and with shocks that follow a Markov chain. Two major bottlenecks create substantial difficulties in solving such dynamic adverse selection problems[3] numerically, namely, (i) the determination and approximation of (possibly) high-dimensional and non-convex equilibrium correspondences—that is, feasible sets of a dynamic program, and (ii) performing dynamic programming on them. Thus, we also have to repeatedly and efficiently approximate functions on irregularly shaped domains. To the best of our knowledge, there exists at present no solution framework in the DI context that can cope with all these issues at once.

Complication (i) is a standard issue in models with repeated agency, since they require full history dependence (see Lambert (1983) and Rogerson (1985)). This, in turn, leads to time-inconsistent dynamic programs. A way of dealing with this issue is to introduce *promised utilities* as a bookkeeping mechanism, which has led to several recursive formulations for various types of hidden information (see, e.g., Spear and Srivastava (1987), Fernandes and Phelan (2000), and Doepke and Townsend (2006)). Once models with history dependence are written in a recursive form, the set of feasible utility vectors is not known in advance and can be of irregular—that is, non-hypercubic geometry. As complication (ii), a *curse of dimensionality* (Bellman, 1961) arises in dynamic adverse selection problems with persistent hidden information, since for every individual type of agent, the dimensionality of the state space increases. Hence, if standard, Cartesian grid-based algorithms are applied in the solution process, the computational effort as well as the storage requirements grow exponentially and render even models of only moderate complexity computationally infeasible. The existing literature, therefore, is limited to at most two-dimensional models (see, e.g., Broer et al. (2017), Doepke and Townsend (2006), and Abraham and Pavoni (2008)), as in them the curse of dimensionality is avoided from the outset, which can be suboptimal with regard to the ideal choice of model.

We propose to tackle (i) by pre-computing the time-invariant feasible set for the continu-

[1] For an incomplete list of references, see, e.g., Spear and Srivastava (1987), Fernandes and Phelan (2000), Cole and Kocherlakota (2001), Werning (2002), Doepke and Townsend (2006), Abraham and Pavoni (2008), and Pavoni et al. (2017).

[2] See, e.g., DeMarzo and Sannikov (2006), Sannikov (2008), Williams (2009, 2011), and He et al. (2017), among others.

[3] Note that while the focus of the work presented lies on solving dynamic adverse selection problems, the method proposed here has a far broader scope: it can also be applied, for example, to moral hazard problems or to dynamic games, where one of the major difficulties also lies in finding the equilibrium sets. For discrete actions and lotteries over payoffs, they are convex. However, for other assumptions they are not, which demands a more general approach such as the one proposed in this paper.

ous state variables by combining ideas from Abreu et al. (1986, 1990) (henceforth APS[4]) with Bayesian Gaussian mixture models (BGMMs) (see, e.g., Rasmussen (2000)). Next, we deal with (ii) by solving the recursively formulated DI problem on an irregularly shaped domain by applying a massively parallelized dynamic programming (DP) algorithm that uses Gaussian process regression (GPR) to approximate high-dimensional value and policy functions (see Scheidegger and Bilionis (2017), and references therein) on the entire feasible set. This combination enables us to analyze DI models that were previously considered to be intractable.

The seminal work by APS introduced a constructive procedure for computing feasible sets. In particular, the authors showed that there exists a monotone set-valued operator whose fixed point is the feasible set, similar to the Bellman operator in dynamic programming. In practical applications we therefore have to repeatedly approximate potentially non-convex equilibrium correspondences. To do so, we apply BGMMs to construct a classifier[5] (see, e.g., Murphy (2012)) to divide the computational domain into a feasible and an infeasible region. This classification then leads to an outer approximation that iteratively shrinks toward the feasible set. We terminate the iteration once two successive classifiers become sufficiently close. Our approach has several desirable features. First, since sampling achieves the approximation of feasible sets through BGMMs, it does not directly suffer from the curse of dimensionality and thus can deal with problems involving many types. Second, the numerical approximation of the feasible set is independent of solving the DI problem and thus does not directly add to the computational complexity. Third, it can approximate both convex and non-convex sets. Thus, the work presented here is a substantial improvement on the previous literature. Judd et al. (2003) and Yeltekin et al. (2017), for example, provide a numerical scheme for determining the feasible sets of discrete state supergames by using polygons. Their approach however relies on the convexification of the payoff set and suffers from the curse of dimensionality. Similarly, Sleet and Yeltekin (2016) provide an extension to this method for the case of continuous state variables, but their extension suffers from the same issues.

After pre-computing the feasible sets, we have to solve the recursively formulated dynamic adverse selection model on the irregularly shaped domains. For this purpose, we apply a massively parallelized discrete-time DP algorithm that uses GPR in order to approximate the value and policy functions. GPR is a form of supervised machine learning (see, e.g., Rasmussen and Williams (2005) and Murphy (2012), and references therein) and has successfully been applied to a variety of applications in data science, engineering, and other fields to perform approximation and classification tasks. In economics, Scheidegger and Bilionis (2017) recently applied Gaussian processes (GPs) to solve very-high-dimensional dynamic stochastic growth models as well as to perform uncertainty quantification. A defining feature of GPs is that they combine the best of two worlds—namely, those of grid-free methods such as Monte Carlo (MC) (see, e.g., Press et al. (2007), and references therein) and of smooth function approximation theory. GPs learn a function based on the observations available at so-called design points, and do so without any geometric restriction. Moreover, since the construction of GP surrogates[6] is achieved by sampling from a domain of interest such as a feasible set, they do not directly suffer from the curse of dimensionality as do Cartesian grid-based algorithms. GPs, therefore, stand in stark contrast to ordinary, grid-based approximation schemes for continuous, high-dimensional state spaces such as Smolyak's method (see, e.g., Malin et al. (2010) and Judd et al. (2014)), adaptive sparse grids (see, e.g., Brumm and Scheidegger (2017)), high-dimensional model reduction (see Eftekhari et al. (2017)), or projection methods (see, e.g., Judd (1992)).

[4]For the remainder of this paper, APS is used to refer both to Abreu et al. (1986, 1990) and to the method introduced by these authors.

[5]Note that in the machine learning literature, classification can—loosely speaking—be considered to be the problem of identifying to which of a set of categories a new data observation belongs.

[6]We use the terms "interpolator" and "surrogate" interchangeably.

Those grid-based approximators are restricted to hyper-rectangular state spaces and thus are not a natural modeling choice in the context of solving dynamic adverse selection models.

To demonstrate the capabilities of the framework proposed in this paper, we solve a dynamic adverse selection model similar to those discussed, for example, in Fernandes and Phelan (2000) and Broer et al. (2017). It consists of a risk-averse agent who has unobserved income shocks, and a risk-neutral planner who wants to provide optimal incentive-compatible insurance against income shocks. The agent reports his income shock to the principal, who then transfers consumption or charges a fee to the agent dependent on the reported shock. Since the principal can only see the reports, the observed shock process is fully history dependent.[7] The reason for this history dependence is that the principal has to condition his actions with respect to the agent's reports and not with respect to the actual shocks. While this model is relatively easy to explain, its dimensionality and complexity can be scaled up in a straightforward and meaningful way, as they just depend linearly on the number of types considered. This feature of the model allows us to focus on the problem of handling irregularly shaped, high-dimensional state spaces in the DI context. Note that much of the existing literature has focused on the stylized analysis of such contracts rather than on their numerical implementation.

The remainder of this paper is organized as follows: In Section 2, we specify the baseline DI environment we are targeting with our framework. In Section 3, we outline the solution method. We first discuss how we construct an APS-style algorithm by using BGMMs; then, we provide a short review of DP and summarize the workings of GP machine learning. Finally, we show how to embed APS and BGMMs into a massively parallel DP algorithm that is based on GPR. In Section 4, we discuss the performance of our method via a variety of illustrative test cases. Section 5 concludes.

2 A baseline dynamic incentive model

To demonstrate the scalability and flexibility of the method we introduce in this paper, we consider an infinitely repeated, dynamic adverse selection problem in discrete time as described in Fernandes and Phelan (2000).[8]

Time is indexed by $t = 1, 2, \ldots \in \mathbb{N}$. In every period t, an agent observes his[9] taste shock $\theta_t \in \Theta$, where Θ is a finite set of cardinality N. He then reports his shock to the principal. Subsequently, the principal offers a contract c_t, which depends on the entire history of reports, to the agent. Since we need to keep track of the history of types (see Fernandes and Phelan (2000)), we define the type history at time t to be the (t)-tuple—that is, an ordered list $h^t = (\theta_0, \theta_1, \ldots, \theta_{t-1})$ for $t \geq 1$. Next, we define the set of possible histories at time t by $H^t = \Theta^t = \Theta \times \Theta \times \ldots \times \Theta$ and set $h^1 \in \Theta$ to be the initial history at time 1. We assume without loss of generality that the initial history is public knowledge. An optimal solution to the problem is a transfer scheme that maximizes the principal's utility while delivering a predetermined lifetime utility to the agent. This scheme will be a sequence of conditional transfers that depend on all past realizations of types. Furthermore, we impose the following assumptions on the primitives of the model:

Assumption 1.

1. *The set of types $\Theta = \{1, \ldots, N\}$ is of cardinality $N \in \mathbb{N}$.*
2. *The set $C \subset \mathbb{R}$ of compensations is a non-empty and compact interval.*

[7] Previous research suggests that private information in the economic environments we are interested in is highly persistent (see, e.g., Meghir and Pistaferri (2004) and Storesletten et al. (2004)).

[8] We follow the formalism of Golosov et al. (2016) to describe the model.

[9] In this paper, for the sake of brevity, both the agent and the principal are male.

3. *The transition probabilities $\pi(\theta_t|\theta_{t-1})$ are defined by a probability matrix $\Pi \in \mathbb{R}^{N\times N}$.*

4. *The principal's utility function is given by $v : C \times \Theta \to \mathbb{R}$ and the agent's utility function by $u : C \times \Theta \to \mathbb{R}$. They are both twice continuously differentiable in C.*

5. *The principal and the agent have a common discount factor $\beta \in (0, 1)$.*[10]

For each period t and history $h^t = (h_0, \ldots, h_{t-1}) \in H^t$ the probability distribution on the set H^t of histories is given by

$$\pi\left((h^t, i)|h^t\right) = \Pi_{\theta_{t-1},i},\ \forall i \in \{1, \ldots, N\} \tag{1}$$

and by recursion for $\tau \geq 1$,

$$\pi\left(\left(h^{t+\tau}, i\right)|h^t\right) = \Pi_{\theta_{t+\tau-1},i} \cdot \pi\left(h^{t+\tau}|h^t\right), \tag{2}$$

where (h^s, i) generally denotes the history h^{s+1} where i has happened at time s, and h^s before that. The principal's compensation strategy is a function of the history of reports up to period t. His strategy, therefore, is a function $c_t : H^t \to C$. We denote the principal's infinite horizon strategies by the sequences of strategy functions $\mathfrak{c} = (c_1, c_2, \ldots)$.

At time t, the agent has to decide what to report to the principal; his strategy is, thus a function on the domain H^t—that is, $a_t : H^t \to \Theta$. A strategy $\mathfrak{a} = (a_t)_{t=1}^{\infty}$ is called incentive compatible iff

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} u\left(c_t(h^t, a_t(h^t, X_t)), X_t\right)\middle| h^1\right] \geq \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} u\left(c_t(h^t, \tilde{a}_t(h^t, X_t)), X_t\right)\middle| h^1\right],\ \forall \tilde{\mathfrak{a}}, \tag{3}$$

where X_t is a random variable with values in Θ and distribution π, and $\tilde{\mathfrak{a}} = (\tilde{a}_t)_{t=1}^{\infty}$ is a feasible strategy. We can get rid of the reporting strategy $\mathfrak{a}$ of the agent by applying the following theorem:

Theorem 1 (Revelation principle)**.** *(Golosov et al., 2016, Th. 1) For every contract $\mathfrak{c}$ and incentive compatible strategy $\hat{\mathfrak{a}}$, there is a $\hat{\mathfrak{c}}$ with the same payoff to the principal such that truth-telling is incentive compatible.*

The revelation principle allows us to only look at compensation schemes that induce truth-telling and thus to define the *dynamic adverse selection problem*. Given an initial history $h^1 = \{\theta_0\}$, the principal faces the following infinite horizon utility maximization problem:

$$\max_{\mathfrak{c}} \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} v\left(c_t(h^t, X_t), X_t\right)\middle| h^1\right] \tag{4}$$

subject to the truth-telling constraint

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} u\left(c_t(h^t, X_t), X_t\right)\middle| h^1\right] \geq \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} u\left(c_t(h^t, a_t(X_t, h^t)), X_t\right)\middle| h^1\right],\ \forall \mathfrak{a} \tag{5}$$

and the reservation utility vector $\tilde{w}$ yields the constraints

$$\tilde{w}_\theta \leq \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^{t-1} u\left(c_t(h^t, X_t), X_t\right)\middle| h^1\right],\ \forall \theta \in \Theta. \tag{6}$$

As a next step, we simplify the incentive constraints by applying a classic theorem.

[10] This last assumption, while being standard, is not necessary for our computational framework.

Theorem 2 (One-shot deviation principle)**.** *(Golosov et al., 2016, Lem. 3) The incentive constraint* (5) *holds iff*

$$\mathbb{E}\left[\sum_{t=1}^{\infty}\beta^{t-1}u\left(c_t(h^t,X_t),X_t\right)\middle|h^1\right]\geq$$
$$\mathbb{E}\left[u\left(c_s(h^s,a(X_s)),X_s\right)+\sum_{t=1,t\neq s}^{\infty}\beta^{t-1}u\left(c_t(h^t,X_t),X_t\right)\middle|h^1\right],\ \forall s\in\mathbb{N},\forall a:\Theta\to\Theta. \tag{7}$$

Note that in the one-shot deviation constraint (see Eq. (7)) we do not allow for all reporting strategies. Instead, only strategies that remain feasible for the agent are admissible. As a concrete example, assume that an agent over-reports his income when filling out his tax declaration. In such a case, he would be charged income tax that he might be unable to pay. All such reports are excluded from Eq. (7).

However, even after all these simplifications the dynamic adverse selection problem is still unsolvable in its present form. To address this, we follow Fernandes and Phelan (2000) and introduce the utility promise w_θ as a continuous state variable.

Theorem 3. *(Golosov et al., 2016, Eqs. (44–47)) There exists $W(\theta)\subset\mathbb{R}^N$, the feasible sets of utility promises for type θ, such that the problem defined by Eqs. 4,6, and 7 can be written recursively as*

$$\begin{aligned}V(w,\theta)=\max_{c,w^+}&\sum_{\tilde\theta\in\Theta}\Pi_{\theta,\tilde\theta}(v(c_{\tilde\theta},\tilde\theta)+\beta V^+(w^+_{\tilde\theta},\tilde\theta))\\ s.t.\ &w_\nu=\sum_{\tilde\theta\in\Theta}\Pi_{\nu,\tilde\theta}(u(c_{\tilde\theta},\tilde\theta)+\beta w^+_{\tilde\theta,\tilde\theta})\ \forall\nu\in\Theta,\\ &u(c_{\tilde\theta},\tilde\theta)+\beta w^+_{\tilde\theta,\tilde\theta}\geq u(c_\nu,\tilde\theta)+\beta w^+_{\nu,\tilde\theta}\ \forall\tilde\theta\in\Theta,\ \forall\nu\in\Theta\setminus\{\tilde\theta\},\\ &c\in[0,\overline{c}]^N,\\ &w^+_{\tilde\theta}\in W(\tilde\theta)\ \textit{with}\ w^+_{\tilde\theta,\nu}=(w^+_{\tilde\theta})_\nu\ \forall\tilde\theta\in\Theta,\ \forall\nu\in\Theta.\end{aligned} \tag{8}$$

Note that we arranged w^+ as an $N\times N$ matrix where the θ-th row of the matrix represents next period's utility promise conditional that θ happens. The first constraint is called promise keeping constraint*. It ensures that the principal follows up on his utility promises w^+_θ. The second constraint is the* truth telling constraint*.*

This measure allows us at the same time to keep track of the full history and to obtain a recursive formulation. The latter point is critical in making the model tractable for the computational method we propose in Sec. 3. Fig. 1 depicts the timeline for the recursively formulated problem. At period t, the principal "learns" last period's report θ and the utility promises w. He then offers the agent a consumption menu c for each possible report, which satisfy the truth-telling and promise-keeping constraints. Next, the agent chooses the truth-telling reporting strategy. Finally, the shock θ_{true} is realized, the agent reports θ_r according to his strategy, and both the principal and the agent receive their respective utilities, $v(c_{\theta_r},\theta_r)$ and $u(c_{\theta_r},\theta_{true})$.

It is not straight forward to see how the recursive problem stated in (8) relates to the original one (see (4),(5), and (6)). To obtain the actual solution, we select a starting utility promise by maximizing over the recursive value function given the constraint that we provide at least the reservation utility. Starting at this utility promise and an initial history, we can now extract the current consumption transfer as well as the next period's utility promises from the recursive problem. In this way, we can iteratively obtain the optimal policies for the

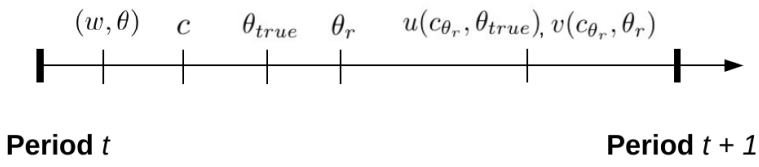

Figure 1: The sequence of events in period t.

non-recursive problem stated in Eqs. (4), (5), and (6). The detailed procedure is summarized in the following Corollary:

Corollary 1. *Let V be the value function of (8), $W(\theta)$ its domain, and c^*, w^* the optimal policies. Assume the initial state is $h^1 = \{\tilde{\theta}\}$. Then the optimal value $\tilde{V}$ of the original problem (4), (5), and (6) is*

$$\tilde{V} = \max_{w \in W(\tilde{\theta})} V(w, \tilde{\theta}) \text{ subject to } w_\theta \geq \tilde{w}_\theta \ \forall \theta. \tag{9}$$

In particular we can find an optimal policy by selecting

$$\hat{w} \in \arg\max_{w \in W(\tilde{\theta})} V(w, \tilde{\theta}) \text{ subject to } w_\theta \geq \tilde{w}_\theta \ \forall \theta, \tag{10}$$

and define

$$\begin{aligned} w_1(h^1) &= \hat{w}, \\ c_1(h^1, \theta) &= c^*_\theta(w_1(h^1), \tilde{\theta}), \ \forall \theta, \\ w_t(h^{t-1}, \theta) &= w^*(w_{t-1}(h^{t-1}), \theta), \ \forall \theta, t > 1, \\ c_t((h^{t-1}, \overline{\theta}), \theta) &= c^*_\theta(w_t(h^{t-1}, \overline{\theta}), \overline{\theta}), \ \forall \theta, \end{aligned} \tag{11}$$

where $(h^{t-1}, \overline{\theta}) = h^t$.

3 Dynamic programming on irregularly shaped domains

Solving dynamic adverse selection models as described in Sec. 2 numerically is a formidable task, since we have to deal with a variety of complex issues at the same time: i) The feasible sets $W(\theta) \subset \mathbb{R}^N$ of utility promises for type θ are not known in advance and have to be determined numerically (see Theorem 3). In most of the interesting cases, such sets have a non-trivial—that is, a non–hypercubic geometry (see, e.g., Fernandes and Phelan (2000), Broer et al. (2017)). ii) For solving a DI problem recursively, for example with value function iteration (VFI; see, e.g., Judd (1998) and Ljungqvist and Sargent (2000)), we need to repeatedly approximate and evaluate high-dimensional functions at arbitrary coordinates within domains of interest—that is to say, the irregularly shaped feasible sets. To meet all these challenging modeling demands we therefore apply set-valued dynamic programming techniques in combination with BGMMs to determine irregularly shaped equilibrium value correspondences. Subsequently, we use a massively parallel VFI algorithm that applies GPR in each iteration step to learn the value function and, if needed, the policy functions on the pre-computed, time-invariant feasible sets globally.[11]

[11] Below, we follow Brumm and Scheidegger (2017) and use the term "global solution" for a solution that is computed using equilibrium conditions at many points in the state space of a dynamic model—in contrast to a

In this section we therefore proceed in four steps. In Sec. 3.1, we characterize the general structure of the models we aim to solve by DP. In Sec. 3.2, we present a novel way of efficiently computing equilibrium correspondences by combining ideas from Abreu et al. (1986, 1990) with BGMMs (see Rasmussen (2000)). Subsequently, we summarize–in Sec. 3.3–how value and policy functions can be approximated by GP machine learning. Sec. 3.4 finally combines all these components into a generic VFI framework for DI problems.

3.1 Background on dynamic programming

Throughout this paper, the abstract class of models we consider are discrete-time, infinite-horizon stochastic optimal decision-making problems. Following Scheidegger and Bilionis (2017), we briefly characterize them here by the subsequent general description: let $x_t \in W \subset \mathbb{R}^N$ denote the state of the economy at time $t \in \mathbb{N}^+$ of dimensionality $N \in \mathbb{N}$. Controls (actions) are represented by a *policy function* $c : W \to \zeta$, where ζ is the space of possible controls. The discrete-time transition function of the economy from one period to the next is given by the distribution of x_{t+1}, which depends on the current state and policies

$$x_{t+1} \sim f\left(\cdot | x_t, c(x_t)\right). \tag{12}$$

The transition function f that stochastically maps a state-action pair to a successor state is assumed to be given, whereas the policy function c needs to be determined from equilibrium or optimality conditions. The standard way to do so is to use DP (see, e.g., Bellman (1961), Stokey et al. (1989), Judd (1998), Ljungqvist and Sargent (2000)), where the task is to find an infinite sequence of *controls* $\{\chi_t\}_{t=0}^{\infty}$ to maximize the *value function*

$$V(x_0) := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t r(x_t, \chi_t)\right] \tag{13}$$

for an initial state $x_0 \in W$, $r(\cdot, \cdot)$ is the so-called *return function*, and $\chi_t \in \Gamma(x_t)$ with $\Gamma(x_t)$ being the set of feasible choices given x_t. The discount factor $\beta \in (0, 1)$ weights future returns. DP seeks a time-invariant policy function c mapping the state x_t into the action χ_t, such that for all $t \in \mathbb{N}$

$$\chi_t \quad = \quad c(x_t) \in \Gamma(x_t), \tag{14}$$

and $\{\chi_t\}_{t=0}^{\infty}$ solves the original problem. The *principle of optimality* states that we can find such a solution by solving the *Bellman equation*

$$V(x) = \max_{\chi} \{r(x, \chi) + \beta \mathbb{E}\left[V(\tilde{x})\right]\}, \tag{15}$$

where the successor state is distributed as $\tilde{x} \sim f\left(\cdot | x, \chi\right)$. The solution is a fixed point of the Bellman operator T, defined by

$$(TV)(x) = \max_{\chi} \{r(x, \chi) + \beta \mathbb{E}\left[V(\tilde{x})\right]\}. \tag{16}$$

Under appropriate conditions (see, e.g., Stokey et al. (1989)) the Bellman operator is a contraction mapping. In this case, iteratively applying T provides a sequence of value functions that converges to a unique fixed point. This procedure is called *value function iteration* (see,

"local solution", which rests on a local approximation around a steady state of the model. For a method that computes such a global solution, we use the term "global solution method". This use of the word "global" is not to be confused with its use in the phrase "global optimization method", which refers to a method that aims to find a global optimum.

e.g. Bertsekas (2000), Judd (1998), or Ljungqvist and Sargent (2000)) and is motivated by this theoretical justification and numerically implements the iterative application of the Bellman operator to successive approximations of the value function. The corresponding DP recursion thus starts from any bounded and continuous guess for the value function, and the solution is approached in the limit as $j \to \infty$ by iterations on

$$V^{j+1}(x) = T(V^j)(x) := \max_{\chi^{j+1}} \{r(x,c) + \beta \mathbb{E}\left[V^j(\tilde{x})\right]\}. \tag{17}$$

In practice, we say that VFI has converged if numerical convergence in some norm, for example

$$\|V^\tau - V^{\tau-1}\|_2 \leq \epsilon, \quad \epsilon \in \mathbb{R}^+, \tag{18}$$

and some at some iteration step τ, is reached. The (approximate) equilibrium value function shall be denoted as $V^* = V^\tau$ and the corresponding policy functions as $\chi^* = \chi^\tau$. From Eq. (17), it becomes apparent that we need to repeatedly approximate and evaluate (potentially multi-dimensional) value functions. An additional complication stems from the fact that domain W for the models we are targeting (cf. Sec. 2) is often highly complex—that is, irregularly shaped and not known a priori (see, e.g., Fernandes and Phelan (2000) and Broer et al. (2017)). We therefore describe in Sec. 3.2 how we determine W numerically, whereas in Sec. 3.3 we show how we approximate value and policy functions on irregularly shaped feasible sets.

3.2 On the iterative approximation of irregularly shaped feasible sets

One major complication we are facing is the fact that the feasible set—that is, the state space for dynamic adverse selection problems is unknown and potentially of irregular geometry. If the problem is simple enough, then it can be possible to find an analytical solution. However, we are usually at most able to give an estimate of the hypercubic domain that contains it, namely–

$$\underline{w}_\theta = \min_{c \in C} \frac{u(c,\theta)}{1-\beta}, \quad \overline{w}_\theta = \max_{c \in C} \frac{u(c,\theta)}{1-\beta}, \tag{19}$$

where $\underline{w}_\theta$ and $\overline{w}_\theta$ are the lower and upper bounds on promised utilities, respectively. Hence, in most of the interesting settings, a numerical procedure for approximating the equilibrium correspondences is required.

One possible way of getting around this issue is to relax the recursive model formulation (see Eq. (8)) by introducing slack variables on the constraints. Whenever they are nonzero, one adds a penalty term to the objective function.[12] Since the penalty quickly becomes large outside the original feasible region, the actual solution can be found by restricting the relaxed problem to the feasible set (see, e.g., Judd et al. (2016), and references therein). The advantage of this procedure is that one can apply highly tuned DP algorithms for hypercubic domains (see, e.g., Brumm and Scheidegger (2017)) to solve DI models. The major disadvantage of this brute-force approach, however, lies in the fact that we need to approximate a computational domain of which large parts might be infeasible, which in turn can result in a massive waste of resources (see, e.g., Scheidegger and Bilionis (2017) for a detailed discussion).

We therefore propose a novel, simulation-based method for determining irregularly shaped feasible sets. This approach will enable us to concentrate the computational resources where they are needed and thus be highly efficient (cf., Secs. 3.4 and 4). To this end, we follow the classical work by APS, who provide set-valued DP techniques for determining feasible sets. In particular, they show that the feasible sets $W(\theta)$ can be found by repeatedly applying a set

[12] Note that in Appendix A, we will apply this procedure in combination with a highly-tuned adaptive sparse grid code (see Brumm and Scheidegger (2017)) to verify the method we propose in this paper.

operator $\mathcal{B}$ to an initial "candidate" set until the resulting sequence of sets converges in some metric. In our case (see Sec. 2), this means that we can define the set-valued operator as

$$\begin{aligned}\mathcal{B}(X) = \Big\{ w \,|\exists (c, w^+) \text{ with } w_\nu = \sum_{\tilde{\theta}\in\Theta} \Pi_{\nu,\tilde{\theta}}(u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}}) \ \forall \nu \in \Theta, \\ u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}} \geq u(c_\nu, \tilde{\theta}) + \beta w^+_{\nu,\tilde{\theta}} \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta \setminus \{\tilde{\theta}\}, \\ c \in [0, \overline{c}]^N, \\ w^+_{\tilde{\theta}} \in X \text{ with } w^+_{\tilde{\theta},\nu} = (w^+_{\tilde{\theta}})_\nu \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta \Big\}, \end{aligned} \tag{20}$$

where $X \subset \mathbb{R}^N$, and where the operator $\mathcal{B}(X)$ contains the constraints of the original problem stated in Eq. (8). This is a monotone operator—that is to say, $\mathcal{B}(X) \subset X$, and its fixed point will be the feasible set for type θ—namely $\mathcal{B}(W(\theta)) = W(\theta)$.

A numerical implementation of the APS method requires the repeated approximation of candidate equilibrium value correspondences—that is, a finite collection of sets. To do so, we propose applying BGMMs[13] to construct a classifier to divide the computational domain into a feasible and an infeasible region. This classification then leads to an outer approximation that iteratively shrinks toward the feasible set. We terminate the iteration once two successive classifiers become sufficiently close.

In practice, we start off the APS iteration by uniformly drawing, for every type, m sample points

$$X_{test} = \left\{ w^{(1)}, \ldots, w^{(m)} \right\} \tag{22}$$

from a hypercube that contains the feasible set (see Eq. (19)) and fit a BGMM to the points $X_{\text{feas}} \subset X_{\text{test}}$ that were deemed to be feasible. We denote the log-likelihood that corresponds to the BGMM as f_θ. Then, we start with the set-valued DP procedure. To do so, we define f^+_θ as the log-likelihood from the BGMM in iteration $i-1$, and denote ℓ^+_θ to be the smallest log-likelihood that corresponded to a feasible point—that is, $\ell^+_\theta = \min\left\{ f^+_\theta(w^{(j)}) \mid w^{(j)} \in (X_{\text{feas}})_\theta \right\}$, and generate another m sample points $\tilde{X}$ from inside the updated feasible set via the respective BGMMs. At a sample point $w^{(k)} \in \tilde{X}$ that satisfies $f^+_\theta(w^{(k)}) \geq \ell^+_\theta$ within an iteration step i, we transform Eq. (20) into an optimization problem[14], namely–

$$\begin{aligned}\varphi = \max_{c, w^+} \sum_{\theta\in\Theta} & -\log(\exp(-\alpha(f^+_\theta(w^+_\theta) - \ell^+_\theta)) + 1)/\alpha + N\log(2)/\alpha, \\ \text{s.t. } & w_\nu = \sum_{\tilde{\theta}\in\Theta} \Pi_{\nu,\tilde{\theta}}(u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}}) \ \forall \nu \in \Theta, \\ & u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}} \geq u(c_\nu, \tilde{\theta}) + \beta w^+_{\nu,\tilde{\theta}} \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta \setminus \{\tilde{\theta}\}, \\ & c \in [0, \overline{c}]^N, \\ & w^+_{\tilde{\theta},\nu} = (w^+_{\tilde{\theta}})_\nu \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta. \end{aligned} \tag{23}$$

[13] Mixture of Gaussians are usually used to approximate probability distributions from observed data. Suppose that we have m data samples $\mathbf{X} = \{\mathbf{x}_i : 1 \leq i \leq m\}$. We then can approximate $\rho_{estimated}$ as a mixture of Gaussians:

$$\rho_{estimated}(\mathbf{x}) = \sum_{l=1}^{L} \pi_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l), \tag{21}$$

where the mean vectors $\boldsymbol{\mu}_l \in \mathbb{R}^N$, the covariance matrices $\boldsymbol{\Sigma}_l \in \mathbb{R}^{N\times N}$, the weights π_l with $\sum_{l=1}^{L} \pi_l = 1$, and the number of components L are fitted to $\mathbf{X}$ (see, e.g., Rasmussen (2000) and Blei and Jordan (2005)).

[14] In the present form, Eq. (20) is numerically un-tractable due to the constraint $w^+_{\tilde{\theta}} \in W(\tilde{\theta})$—there is no explicit description of the set $W(\theta)$ that is given by inequalities. To circumvent this issue, we get rid of this constraint by introducing a penalty function p. For any point in the feasible set, this function p will be approximately 0 and will tend to minus infinity when we leave the boundary of the set (see Eq. (23)).

Data: For all θ: initial set of points $\{w^{(1)}, \ldots, w^{(m)}\}$, uniformly drawn from a domain that contains the feasible set (see Eq. (19)). Approximation accuracy $\bar{\epsilon}$.

Result: The approximate feasible sets $W(\theta)$, determined by the log-likelihood f_θ and cut-off ℓ_θ that define the classifier C_θ, where $C_\theta(w^{(j)}) = 1$ iff $f_\theta(w^{(j)}) \geq \ell_\theta$.

Fit BGMM to X_{feas} for all θ to get f_θ and $\ell_\theta = \min\left\{f_\theta(w^{(j)}) \mid w^{(j)} \in (X_{\text{feas}})_\theta\right\}$.
$f_\theta^+ = f_\theta$.
$\ell_\theta^+ = \ell_\theta$.
while $\epsilon > \bar{\epsilon}$ **do**
  for $\theta \in \Theta$ **do**
    Generate m sample points $\tilde{X}$ from the most recent "candidate" feasible set via last iteration's BGMM and set $X = \{w^{(k)} \in \tilde{X} : f_\theta^+(x) \geq \ell_\theta^+\}$.
    Set $n = |X|$.
    Set $(X_{\text{feas}})_\theta = \emptyset$.
    while $|(X_{feas})_\theta| < n$ **do**
      Solve optimization problem given by Eq. (23) at point $w^{(k)} \in X$ and get objective φ.
      if $\varphi \geq 0$ **then**
        $(X_{\text{feas}})_\theta = (X_{\text{feas}})_\theta \cup \{w^{(j)}\}$.
      end
    end
    Fit a BGMM to $(X_{\text{feas}})_\theta$ to obtain f_θ and $l_\theta = \min\left\{f_\theta(w) \mid w^{(j)} \in (X_{feas})_\theta\right\}$.
  end
  $f_\theta^+ = f_\theta$.
  $\ell_\theta^+ = \ell_\theta$.
  Compute the average L_1-error ϵ.
end

Algorithm 1: Overview of the critical steps for computing the equilibrium correspondences.

The objective function $F(w_\theta^+, \theta, f_\theta^+, \ell_\theta^+)$ of the constrained optimization problem stated above in Eq. (23) is a smooth relaxation of a classifier that returns approximately 0 for feasible points and a large negative number for infeasible ones. Furthermore, the constraints are the same ones as those in the description of the operator $\mathcal{B}$ in Eq. (20). The parameter α controls how strict the relaxation is: the smaller α is, the quicker the objective in Eq. (23) will go to $-\infty$ when leaving the feasible set.[15] If φ is above 0, we label the point $w^{(j)}$ as feasible and add it to a set of feasible points X_{feas}. We repeat this until the set X_{feas} has sufficiently large cardinality.[16] Subsequently, we fit another BGMM to X_{feas} to obtain updated values for f_θ and ℓ_θ. This procedure is repeated until convergence. Alg. 1 summarizes the detailed steps of the set-valued DP for determining the feasible sets $W(\theta)$ in a more formal way.

Note that it is non-trivial to measure whether the set-valued DP procedure has converged. To this end, we propose the following procedure: In every iteration step I and every state θ, we construct a binary classifier $C_{I,\theta}$ (see, e.g., Murphy (2012)) by labelling infeasible sample points as 0, and feasible ones by 1, and compute the average L_1-error across subsequent candidate equilibrium correspondences. In practice, this amounts to drawing a uniform sample of points from a hypercube that contains the feasible set. Subsequently, we use the classifiers from steps

[15] In practical applications (see, e.g., Sec. 4.1), we set $\alpha = 0.1$.
[16] Note that in our computations (see Sec. 4), $|(X_{feas})_\theta| \approx 1,000$ led to satisfactory results.

$I-1$ and I to find the label of each sample point. We then determine the percentage of points that get different labels from the classifiers at $I-1$ and I, resulting in an approximation of the error.

The approach presented here has three highly desirable features that yield a substantial improvement over the previous literature. First, since sampling achieves the approximation of feasible sets through BGMMs, it does not directly suffer from the curse of dimensionality and thus can deal with problems involving many types of agents. Second, our scheme of approximating equilibrium correspondences is independent of solving the DI problem and thus does not directly add to the computational complexity of solving the actual model. Finally, it can approximate both convex and non-convex sets.

In contrast, Judd et al. (2003) and Yeltekin et al. (2017), for example, approximate feasible sets by linear hyperplanes. Their approach is limited to models with discrete states and convex sets. Moreover, it suffers from the curse of dimensionality. Thus, it is not useful for models with many types and potentially non-convex, state spaces. Abreu and Sannikov (2014) provide a method for computing feasible sets in discrete state, two-player games that is restricted to convex sets and also suffers from the curse of dimensionality. Sleet and Yeltekin (2016) provide an extension to Judd et al. (2003) for the case of continuous state variables, but their method is also restricted to convex sets in low-dimensional spaces.

Once the set-valued DP procedure for finding the equilibrium correspondences has converged, we can use the equilibrium BGMMs to generate training data from within these feasible sets $W(\theta)$ to train the GPs (see Secs. 3.3 and 3.4). This focuses the computational resources where needed. Note that for the same reasons as in Eq. (23), we have to restate the original recursive problem (see Eq. (8)) and replace the constraints on the future utility promises with a penalty F, resulting in

$$
\begin{aligned}
V(w,\theta) = \max_{c,w^+} & \sum_{\tilde{\theta}\in\Theta} \Pi_{\theta,\tilde{\theta}}(v(c_{\tilde{\theta}},\tilde{\theta}) + \beta V^+(w^+_{\tilde{\theta}},\tilde{\theta})) + F(w^+_{\tilde{\theta}},\tilde{\theta},f_{\tilde{\theta}},\ell_{\tilde{\theta}})) \qquad (24)\\
\text{s.t. } & w_\nu = \sum_{\tilde{\theta}\in\Theta} \Pi_{\nu,\tilde{\theta}}(u(c_{\tilde{\theta}},\tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}}) \;\forall \nu\in\Theta,\\
& u(c_{\tilde{\theta}},\tilde{\theta}) + \beta w^+_{\tilde{\theta},\tilde{\theta}} \geq u(c_\nu,\tilde{\theta}) + \beta w^+_{\nu,\tilde{\theta}} \;\forall\tilde{\theta}\in\Theta,\; \forall\nu\in\Theta\setminus\{\tilde{\theta}\},\\
& c\in[0,\overline{c}]^N,\\
& w^+_{\tilde{\theta},\nu} = (w^+_{\tilde{\theta}})_\nu \;\forall\tilde{\theta}\in\Theta,\; \forall\nu\in\Theta,
\end{aligned}
$$

where F is the objective function from the constrained optimization problem stated in Eq. (23). This penalty will be close to zero for $f_\theta \geq \ell_\theta$, but will diverge to $-\infty$ for $f_\theta < \ell_\theta$. Moreover, it is smooth. Hence, we can easily use it in the optimization problem.

3.3 Gaussian process regression

In the following, we provide a very brief introduction to GPR based on Rasmussen and Williams (2005) and Scheidegger and Bilionis (2017), with references therein. GPR is a nonparametric regression method from supervised machine learning, and addresses the problem of learning input–output mappings from observed data—the so-called training set. Observations can for example stem from a computer code (as in our case) or from empirical experiments. More abstractly, given a data set $\mathcal{D} = \{(\mathbf{x}^{(i)}, t^{(i)}) \,|\, i = 1, ..., \mu\}$ consisting of μ input vectors $\mathbf{x}^{(i)} \in W \subset \mathbb{R}^N$ and corresponding, potentially noisy, observations $t^{(i)} = V(\mathbf{x}^{(i)}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, we want to deduce a model of the unknown function V that generated the data such that we then can make predictions for new inputs $\mathbf{x}^*$ that we have not seen in the training set. In the literature, the matrix

$$
\mathbf{X} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\mu)}\right\} \qquad (25)
$$

is commonly referred to as *training inputs*, whereas

$$\mathbf{t} = \left\{ t^{(1)}, \ldots, t^{(\mu)} \right\} \tag{26}$$

is the vector of the corresponding *training targets* (observations). To enable predictions based on information contained in $\mathcal{D}$, we must make assumptions about the characteristics of the underlying functions, as GPR is a Bayesian regression method. We start by defining a probability measure on the function space, where $V(\cdot)$ lives corresponding to our beliefs. Before seeing any data, we model our state of knowledge about $V(\cdot)$ by assigning a GP prior to it. We say that $V(\cdot)$ is a GP with *mean function* $m(\cdot; \boldsymbol{\phi})$ and *covariance function* $k(\cdot, \cdot; \boldsymbol{\phi})$, and write

$$V(\cdot)|\boldsymbol{\phi} \sim \mathrm{GP}(V(\cdot)|m(\cdot; \boldsymbol{\phi}), k(\cdot, \cdot; \boldsymbol{\phi})), \tag{27}$$

where $\boldsymbol{\phi} \in \boldsymbol{\Theta} \subset \mathbb{R}^{d_\theta}$ and $d_\theta \in \mathbb{N}$ are the so-called *hyper-parameters* of the model. The prior beliefs about the response are reflected in our choice of the mean and covariance functions. The prior mean function is required to model any general trends of the response surface and can have any functional form.[17] The covariance function, also known as the *covariance kernel*, is the most important part of GPR: it defines a measure of similarity on the input space. That is to say, given two input points, their covariance models how close we expect the corresponding outputs to be. A valid choice for a covariance function must be positive semi-definite and symmetric. A very popular covariance function is the *square exponential* (SE)

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi}) = s^2 \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \frac{(x_i - x_i')^2}{\ell_i^2} \right\}, \tag{28}$$

where $\boldsymbol{\phi} = \{s, \ell_1, \ldots, \ell_N\}$, with $s > 0$ being the variability of the latent function V, and $\ell_i > 0$ the characteristic lengthscale of the i-th input.[18] We will use the SE kernel in all our numerical experiments below (see Sec. 4).

Given an arbitrary set of training inputs $\mathbf{X}$, Eq. (27) induces a Gaussian prior on the corresponding response outputs:

$$\mathbf{V} = \left\{ V\left(\mathbf{x}^{(1)}\right), \ldots, V\left(\mathbf{x}^{(\mu)}\right) \right\}. \tag{29}$$

In particular, $\mathbf{V}$ is distributed as

$$\mathbf{V}|\mathbf{X}, \boldsymbol{\phi} \sim \mathcal{N}\left(\mathbf{V}|\mathbf{m}, \mathbf{K}\right), \tag{30}$$

where $\mathcal{N}\left(\cdot|\mathbf{m}, \mathbf{K}\right)$ is the PDF of a multivariate Gaussian random variable with $\mathbf{m} := \mathbf{m}(\mathbf{X}; \boldsymbol{\phi}) \in \mathbb{R}^\mu$ being the mean function evaluated at all points in $\mathbf{X}$, and $\mathbf{K} \in \mathbb{R}^{\mu \times \mu}$ is the covariance matrix with $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\phi})$ (see, e.g., Eq. (28)).

In the Bayesian framework we are operating in, we have to model explicitly the measurement process that gives rise to the observations $\mathbf{t}$. We do so by assuming that measurements are independent of one another, and that they are normally distributed about $V(\cdot)$ with variance s_n^2:

$$t^{(i)}|V\left(\mathbf{x}^{(i)}\right), s_n \sim \mathcal{N}\left(t^{(i)}\middle|V\left(\mathbf{x}^{(i)}\right), s_n^2\right), \tag{31}$$

where $s_n > 0$ is an additional hyper-parameter that must be determined from the training targets. Using the independence of the observations, we get

$$\mathbf{t}|\mathbf{V}, s_n \sim \mathcal{N}\left(\mathbf{t}\middle|\mathbf{V}, s_n^2 \mathbf{I}_N\right). \tag{32}$$

[17] We set the prior mean to 0 throughout this paper.

[18] Note that the hyper-parameters of the covariance function are typically estimated by maximizing the likelihood, a topic that is beyond the scope of the present work. For more details, see Scheidegger and Bilionis (2017), and references therein.

In direct consequence, the *likelihood* of the observations is, given the inputs,

$$\mathbf{t}|\mathbf{X}, \boldsymbol{\phi}, s_n \sim \mathcal{N}\left(\mathbf{t}\middle|\mathbf{m}, \mathbf{K} + s_n^2 \mathbf{I}_N\right). \tag{33}$$

Bayes's rule combines the prior GP (see Eq. (27)) with the likelihood (see Eq. (33)) and yields the *posterior* GP

$$V(\cdot)|\mathbf{X}, \mathbf{t}, \boldsymbol{\phi}, s_n \sim \mathrm{GP}\left(V(\cdot)\middle|\tilde{m}(\cdot), \tilde{k}(\cdot,\cdot)\right), \tag{34}$$

where the *posterior* mean and covariance functions are given by

$$\tilde{m}(\mathbf{x}) := \tilde{m}(\mathbf{x}; \boldsymbol{\phi}) = m(\mathbf{x}; \boldsymbol{\phi}) + \mathbf{K}(\mathbf{x}, \mathbf{X}; \boldsymbol{\phi}) \left(\mathbf{K} + s_n^2 \mathbf{I}_N\right)^{-1} (\mathbf{t} - \mathbf{m}) \tag{35}$$

and

$$\begin{aligned} \tilde{k}(\mathbf{x}, \mathbf{x}') &:= \tilde{k}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi}, s_n) \\ &= k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi}) - \mathbf{K}(\mathbf{x}, \mathbf{X}; \boldsymbol{\phi}) \left(\mathbf{K} + s_n^2 \mathbf{I}_N\right)^{-1} \mathbf{K}(\mathbf{X}, \mathbf{x}; \boldsymbol{\phi}), \end{aligned} \tag{36}$$

respectively. In order to carry out interpolation tasks when performing VFI (see Secs. 3.4 and 4), one has to operate with the predictive (marginal) distribution of the function value $V(\mathbf{x}^*)$ for a single test input $\mathbf{x}^*$ conditional on the hyper-parameters $\boldsymbol{\phi}$ and s_n—namely,

$$V(\mathbf{x}^*)|\mathbf{X}, \mathbf{t}, \boldsymbol{\phi}, s_n \sim \mathcal{N}\left(V(\mathbf{x}^*)|\tilde{m}(\mathbf{x}^*), \tilde{\sigma}(\mathbf{x}^*)\right), \tag{37}$$

where $\tilde{m}(\mathbf{x}^*) = \tilde{m}(\mathbf{x}^*; \boldsymbol{\phi})$ is the *predictive mean* given by Eq. (35), and $\tilde{\sigma}^2(\mathbf{x}^*) := \tilde{k}(\mathbf{x}^*, \mathbf{x}^*; \boldsymbol{\phi}, s_n)$ is the *predictive variance*. The predictive mean can be used as a point-wise surrogate of the response surface—that is, the interpolation value.

3.4 A solution algorithm for dynamic incentive problems

Given the DP procedure described in Sec. 3.1 (see Eqs. (16) and (17)), we now outline how to solve a dynamic adverse selection model (cf. Sec. 2) recursively by combining the pre-computation of feasible sets with VFI and GPR.

The GP VFI algorithm that we propose for computing the optimal decision rules proceeds as follows: For every discrete state present in the model (that is, every individual type θ), we first determine the static—that is, time-invariant feasible set $W(\theta)$ (see Sec. 3.2) by BGMMs, from which we then later can sample training inputs for the GPR. Next, we make an initial guess for a value function $V_0(\cdot,\cdot)$ from which we can instantiate the VFI procedure. Then, we generate at every iteration step $j+1$ of the VFI and for every discrete state θ a set of μ training inputs from within the feasible set $W(\theta)$–namely, $\mathbf{x}_{1:\mu}^{j+1}$, on which we evaluate the Bellman operator (see Eqs. 16 and 24)[19]

$$\mathbf{t}_{1:\mu}^{j+1} = \{\mathbf{t}_1^{j+1}, \ldots, \mathbf{t}_\mu^{j+1}\}, \tag{38}$$

where

$$\mathbf{t}_i^{j+1} = T(V^j)(\mathbf{x}_{1:\mu}^{j+1}, \theta^{j+1}). \tag{39}$$

We use this training data to learn the surrogate of the updated value function for type θ. Note that every individual evaluation of the Bellman operator is carried out by using the predictive mean of $V^j(\cdot,\cdot)$ as the interpolator. The VFI procedure is continued until numerical convergence is reached (cf. Eq. (18)). Alg. 2 summarizes the detailed steps of the GP VFI on multidimensional, irregularly shaped feasible sets $W(\theta)$ in a more formal way. Note that at convergence, one can not only learn the approximate value functions $V^*(\cdot,\cdot) = V^\tau(\cdot,\cdot)$ but also, if desired, the equilibrium policy functions $\chi^*(\cdot,\cdot)$, using an individual GP per policy.

[19] At every single training point, the individual optimization problems—given by Eq. (24) in our application—are solved with `Ipopt` (see Waechter and Biegler (2006)), which is a high-quality open-source software for solving nonlinear programs (`http://www.coin-or.org/Ipopt/`).

Data: Initial guess $V_0(\cdot,\cdot)$ for the value function of every type θ. Approximation accuracy $\bar{\epsilon}$.

Result: The M (approximate) equilibrium policy functions $\chi^*(\cdot,\cdot)$ and the corresponding value functions $V^*(\cdot,\cdot)$ for every type $\theta \in \mathbf{\Theta}$, where $|\mathbf{\Theta}| = N$.

Determine for every type θ the respective feasible set $W(\theta)$ by iterating on Eq. (23).
Set iteration step $j = 0$.
while $\epsilon > \bar{\epsilon}$ **do**
  for $\theta^{j+1} \in \mathbf{\Theta}$ **do**
    Generate μ training inputs $\mathbf{X} = \{\mathbf{x}_i^{j+1} : 1 \leq i \leq \mu\} \in W(\theta)$.
    for $\mathbf{x}_i^{j+1} \in \mathbf{X}$ **do**
      Evaluate the Bellman operator $T(V^j)(\mathbf{x}_i^{j+1}, \theta^{j+1})$ (see. Eq. (24)).
      Set the training targets for the value function: $t_i = T(V^j)(\mathbf{x}_i^{j+1}, \theta^{j+1})$.
    end
    Set $\mathbf{t} = \{t_i : 1 \leq i \leq \mu\}$.
    Given $\{\mathbf{X}, \mathbf{t}\}$, learn the surrogate $V^{j+1}(\cdot, \theta^{j+1})$.
    Compute an error measure, e.g.: $\epsilon_\theta = \|V^{j+1}(\cdot, \theta^{j+1}) - V^j(\cdot, \theta^{j+1})\|_2$.
  end
  Set $j = j + 1$. $\epsilon = \max(\epsilon_{\theta_1}, ..., \epsilon_{\theta_D})$
end
$\tau = j - 1$
$V^*(\cdot,\cdot) = V^\tau(\cdot,\cdot)$.
$\chi^*(\cdot,\cdot) = \{\chi_1(\cdot,\cdot), \cdots, \chi_M(\cdot,\cdot)\}$.

Algorithm 2: Overview of the critical steps of the VFI algorithm that operates on equilibrium correspondences $W(\cdot)$.

Note that one of the defining features of GPR is that it is a grid-free method of constructing a surrogate—that is, it allows the modeler to closely steer the content of the training set $\{\mathbf{X}, \mathbf{t}\}$ (see Sec. 3.3) and thus to construct surrogates on irregularly shaped geometries. This has two significant practical advantages when addressing DI models numerically. First, if an individual optimization problem at some particular point $\mathbf{x_i}$ does not converge, one does not need to deal with tuning the optimizer until it converges at this location of the state space. Instead, this training input (and the corresponding, nonsensical training target) can be discarded—that is to say, it is not added to the training set. This is in stark contrast to grid-based methods such as "Smolyak" (see, e.g., Krueger and Kubler (2004) and Judd et al. (2014)) or "adaptive sparse grids" (see, e.g., Brumm and Scheidegger (2017) and Brumm et al. (2015)), where the construction of the surrogate breaks down if not every optimization problem required by the algorithm can be solved. Second, computing solutions solely on a domain of relevance—that is, $W(\theta)$, allows one to carry out VFI on complex, high-dimensional geometries without suffering from massive inefficiencies, as the computational resources are concentrated where needed. Particularly in high-dimensional settings, this can potentially speed up the time-to-solution process by orders of magnitude, as the feasible set might have a negligibly small volume compared to the computational domain that standard approximation methods require (see Scheidegger and Bilionis (2017) for more details). Finally, note that to solve "large" problems in a reasonably short time, we make use of parallel computation. For the details of the parallelization, see Scheidegger and Bilionis (2017).

Parameter	Value
β	0.9
h_{high}	0.35
h_{low}	0.1
$[\underline{c}, \overline{c}]$	$[0, 1]$

Table 1: Parameterization of the privately observed endowment model by Fernandes and Phelan (2000).

4 Numerical experiments

To demonstrate the broad applicability and versatility of the framework introduced in this paper, we solve three distinct versions of the dynamic adverse selection environment outlined in Sec. 2. In Sec. 4.1, we solve the example by Fernandes and Phelan (2000) as a basic verification test for our method. In Sec. 4.2, we extend this model to alternative preferences. Third, we solve in Sec. 4.3 the baseline model by Fernandes and Phelan (2000) for increasingly many types θ to show that we can handle dynamic adverse selection problems with state spaces of irregular geometry and a dimensionality larger than two, which is the standard in the previous literature.

4.1 Baseline model

To gain a systematic understanding of how our framework behaves in real applications, we apply it to a privately observed endowment model by Fernandes and Phelan (2000), for which numerical solutions are known. First, we briefly summarize the model and its parameterization. Second, we report on the performance of our proposed set-valued DP method. Finally, we discuss the solution to the DI problem that was obtained by performing VFI on the pre-computed, irregularly shaped state spaces.

We consider an environment that consists of a risk-neutral principal who *minimizes* his cost, and a risk-averse agent. We choose the agent's utility over consumption to be

$$u(c) = \sqrt{c}, \tag{40}$$

where c is restricted to the finite range $[\underline{c}, \overline{c}]$. There are two types in the model—namely a "low" (state 1) and "high" (state 2) one. The respective endowments are given by h_{low} and h_{high}. The agent receives a shock in each period and then reports it to the principal. The agent learns his private type in each period and then reports it to the principal. Subsequently, the principal then transfers consumption to agent dependent on what the agent reported (see Fig. 1). Since the problem depends on the full history, we use the recursive reformulation that we introduced in Theorem 3. The resulting state space of promised utilities in this model is 2-dimensional. Furthermore, we follow Fernandes and Phelan (2000) and assume that the agent cannot claim to be a higher type than he is. In particular, if the principal charges more than the lower type can afford, then we have to prevent the agent from over-reporting his type (see Sec. 2). The Markov process governing the endowments in the model specification here is such that the agent has a 90 percent chance of receiving the endowment he received in the previous period. The remaining parametrization is reported in Tab. 1.

In Fig. 2, we display approximations of the equilibrium value correspondences for the low state at various iteration steps when performing set-valued DP with BGMMs (see Alg. 1). In particular, we show the candidate points that were generated from within the current approximation of the feasible set as well as the sample points that are deemed feasible in the

respective APS iteration step. This sequence of figures indicates that our proposed method—that is, to merge the set-valued DP methods with BGMMs, leads to converging results. In

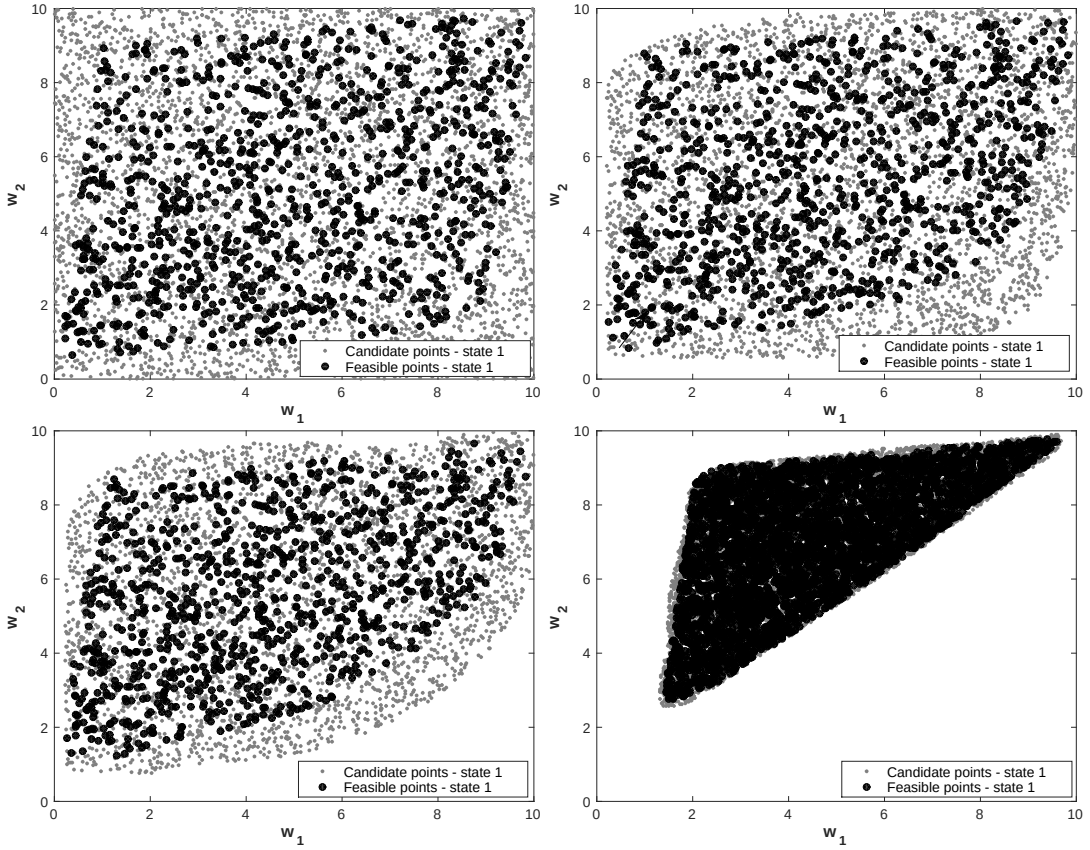

Figure 2: The above four panels display the convergence of the (candidate) feasible set for state 1 as a function of w_1 and w_2 (see Alg. 1). The top-left panel shows the first step in the set-valued DP procedure; the top-right shows step 2. The lower-left panel is a result from step 15, and the lower-right is based on iteration 35.

Fig. 3, we show the feasible sets for both the low and the high state once the set-valued DP iteration has converged—that is, after about 50 iteration steps, where the approximation error for the sets reaches $\mathcal{O}(10^{-3})$ percent.[20] Having (approximately) determined the equilibrium value correspondences for the two states, we turn our attention now to the recursive solution of the model. Fig. 4 depicts the decreasing, aggregated L_1- and L_2-errors (see Eq. (18)) for the value functions when performing VFI with GPs on the feasible sets (see Alg. 2). The graph indicates that our proposed solution framework—that is, the combination of APS, BGMMs, and GPR—can successfully and efficiently solve DI problems.[21] In Fig. 5, we show the value functions for the low and high states at convergence.

To gain some economic insights into the optimal contracts computed, we now carry out a collection of simulations and impulse–response experiments. In the left panel of Fig. 6, we show simulation patterns for the principal's optimal value in the low (V_1) and high (V_2) states. The simulation was launched from the optimal value in the promised utility space—that is,

[20] We cross-validate the findings here by an alternative method that is based on adaptive sparse grids. The affirmative results are summarized in Appendix A.

[21] In this application, generating 100 observations per state and iteration step from the approximate equilibrium correspondences to train the GPs led to satisfactory results.

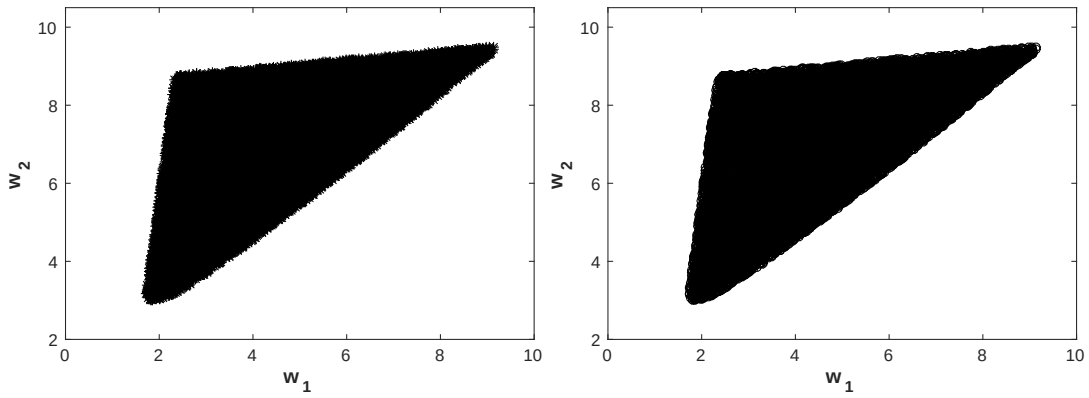

Figure 3: The left panel shows the approximate feasible set for the low state at convergence as a function of w_1 and w_2. The right panel shows the respective set for the high state.

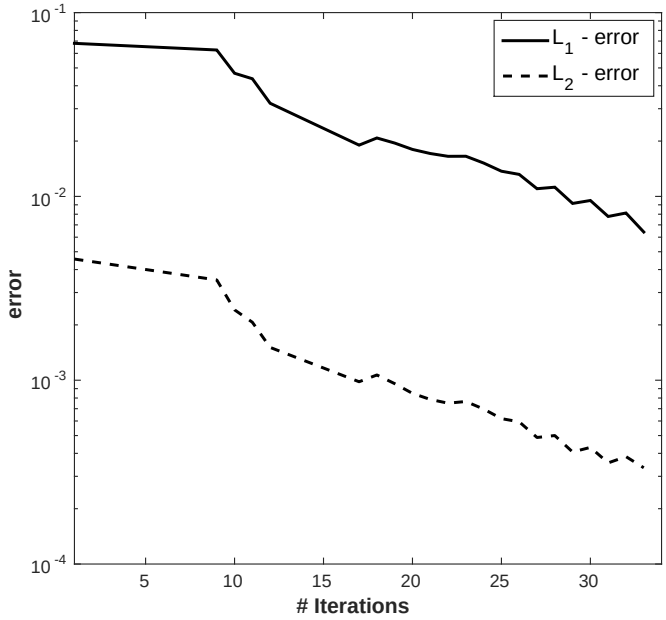

Figure 4: Aggregated, decreasing L_1- and L_2-errors for the 2-dimensional DI model that was solved by training GP surrogates with 100 observations that were generated from within the feasible sets. The errors were computed by randomly drawing $1,000$ points from the feasible sets.

at $w_k^* \in \text{argmin} V_{k=1,2}$. Furthermore, we kept the shocks constant in the respective state. We find that in the high state, the optimal value is increasing, whereas in the low state, the value remains constant. Analogously, the right panel of Fig. 6 displays the utility promise in state i to type i. Fig. 7 displays the corresponding utility transfer u in state i to type i. The left and right panels of Fig. 8 depict the simulation paths within the respective feasible sets. In the low state, we almost start at the steady state, whereas in the high state, we traverse almost the entire set. The reason for this is that there is no incentive constraint for the low type (cf. footnote 3 in Fernandes and Phelan (2000)).

Next, we show several impulse–response experiments on the equilibrium policies. Starting from the steady state[22], we induce the complementary state, and then return to the original

[22]We define the "steady state" as the location in the state space where the simulation process becomes

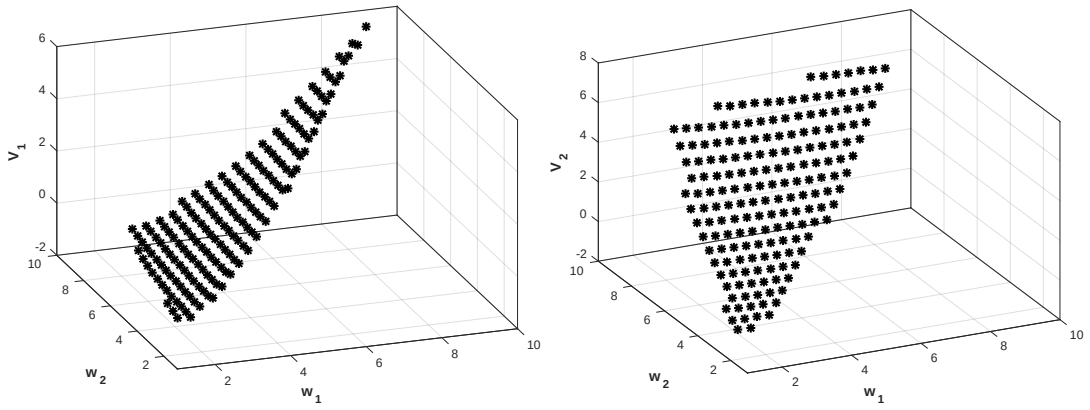

Figure 5: Left panel—value function of the cost-minimization problem in the low state (V_1) as a function of w_1 and w_2. Right panel—value function of the high state (V_2) as a function of w_1 and w_2. In both panels, we evaluated the respective value functions at 180 equally spaced points for illustrative purposes.

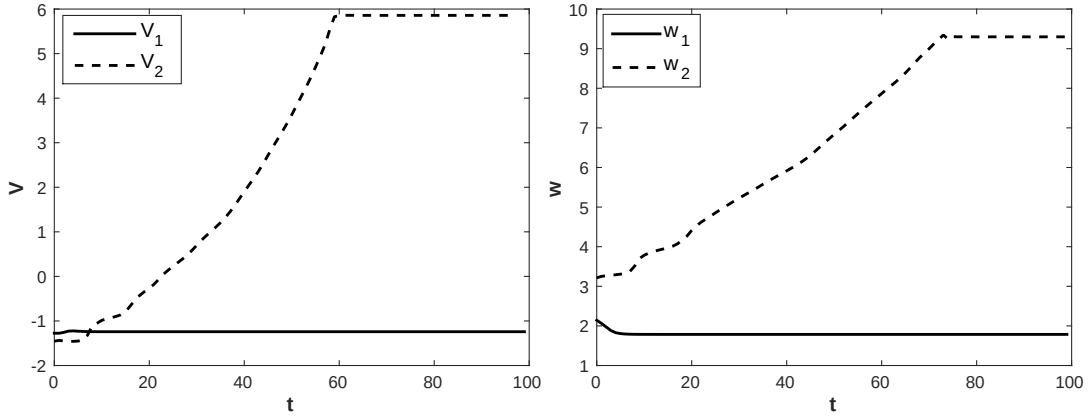

Figure 6: Left panel—optimal value of the cost-minimization problem at simulation step t in states 1 (V_1) and 2 (V_2). Right panel—promised utility in the low (w_1) and high states (w_2) as a function of the simulation step t.

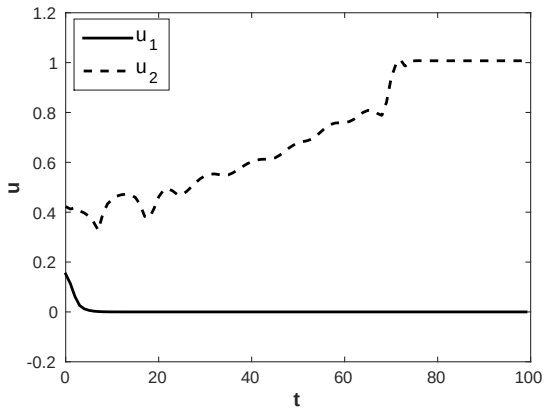

Figure 7: Utility transfer u in states 1 (u_1) and 2 (u_2) to type 1 and 2 at simulation step t.

stationary (see Fig. 8).

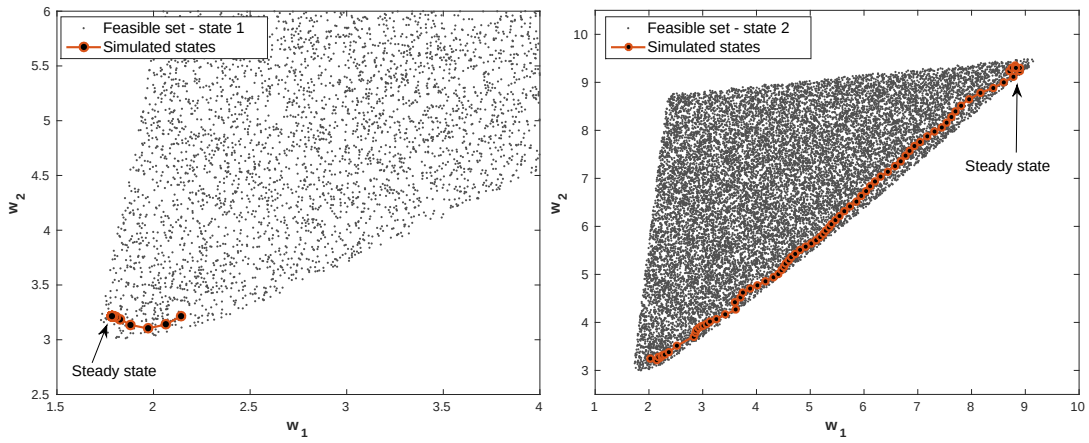

Figure 8: Left panel—simulation path in the feasible set for the low state, state 1. Right panel—the same, for the high state, state 2.

one. The left panel of Fig. 9 displays how the promised utility in the low state reacts to this shock, whereas the right panel shows the same for the utility transfer. We can see that the perturbed quantities quickly return to the steady state, since the discount factor $\beta = 0.9$ is relatively small (see Tab. 1). Fig. 10 shows what happens if we change the type for one period,

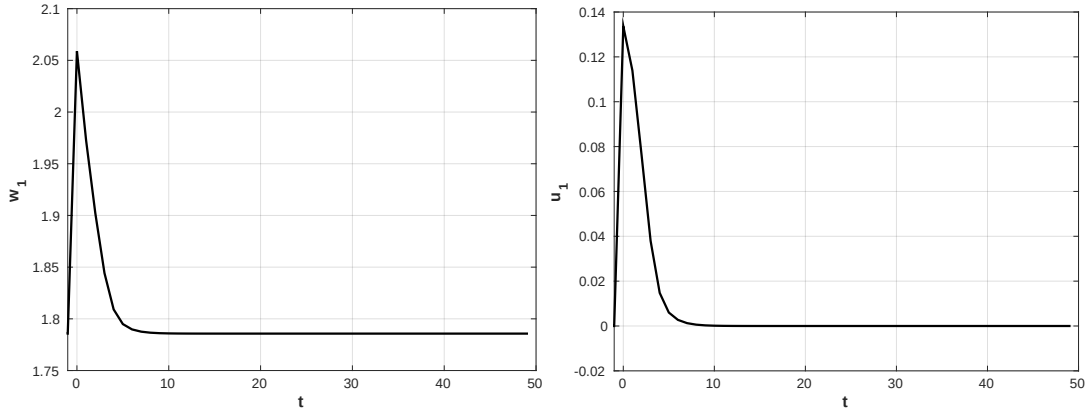

Figure 9: Left panel—behavior of promised utility w_1 if we change the state from 1 to 2 for one period while being in the steady state. Right panel—similar to the left panel but for utility transfer u_1. Note that we induce state 2 at $t = -1$.

while being in the steady state. As shown in this figure, the utility promises return after a few steps back to the steady state for both the low and high types.

Summarizing, we note that the results shown in this section replicate those reported in Fernandes and Phelan (2000).

4.2 Alternative preferences

To show that our framework is capable of operating on a broad variety of preferences, we turn our attention now to a model specification that is slightly different to the one considered in Sec. 4.1. In particular, the risk-averse agent's utility over consumption is now given by

$$u(c) = \frac{(c + \underline{c})^{1-\gamma}}{1-\gamma}. \tag{41}$$

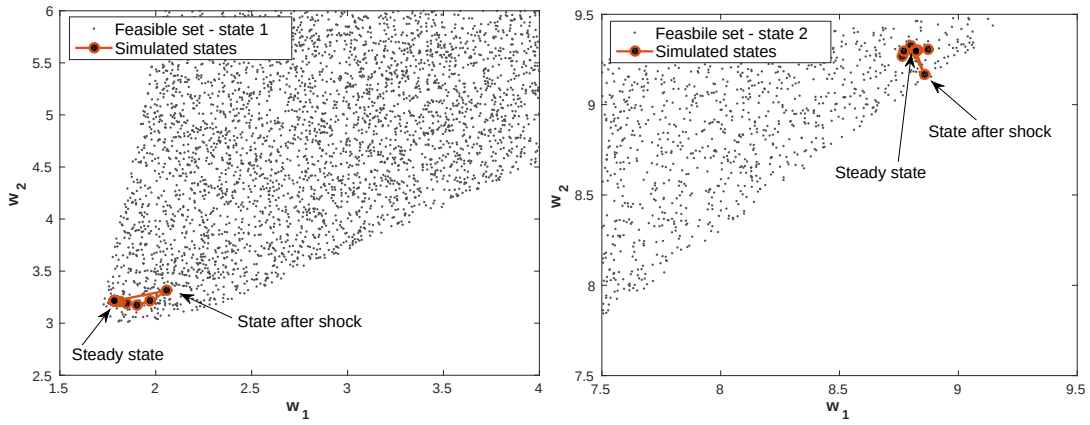

Figure 10: Path of the utility promises after a shock to the steady state has occurred, depicted in the left panel for state 1 and in the right panel for state 2. In both panels, the steady state as well as the utility promises right after the shock are labeled by arrows.

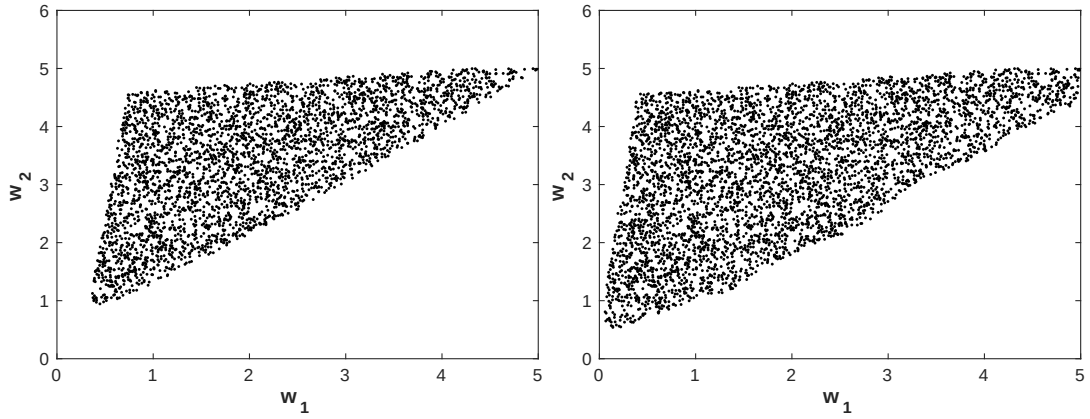

Figure 11: The left panel shows the approximate feasible set for the low state at convergence as a function of w_1 and w_2. The right panel shows the respective set for the high state.

As a concrete example, we choose $\gamma = 2$. Moreover, to ensure that the lower bound on utility at $c = 0$ is 0, we set

$$u(c) = -\frac{1}{c + \underline{c}} + \frac{1}{\underline{c}}, \tag{42}$$

with $\underline{c} = 1$. We keep all other parameters as listed in Tab. 1. Since the upper bound on consumption in this example is 1, the upper bound on utility in turn is given by

$$u(1)/(1 - \beta) = 0.5/0.1 = 5. \tag{43}$$

In Fig. 11, we show approximations for the feasible sets for both the low and the high state once the set-valued DP iteration with BGMMs has converged (see Alg. 1). Comparing Fig. 11 with Fig. 3—that is, the feasible sets for the two respective utility functions, we can see that the sets for both the low and high types are of similar shape, whereas the ranges are substantially different: the preferences in this example yield substantially smaller equilibrium value correspondences.

As in Sec. 4.1, we now carry out simulations and impulse–response experiments to gain insights into the optimal contracts we compute in this section. The simulations were again

launched from the optimal value in the promised utility space. Furthermore, we kept the shocks constant in the respective state.

The left and the right panels of Fig. 12 show the simulation path within the respective equilibrium value correspondences. In line with Sec. 4.1, we find that in the low state we almost start at the steady state, whereas in the high state we traverse a large fraction of the feasible set. The reason for this is again that in the present model setting, there is no incentive constraint for the low type (cf. footnote 3 in Fernandes and Phelan, 2000). Next, we show

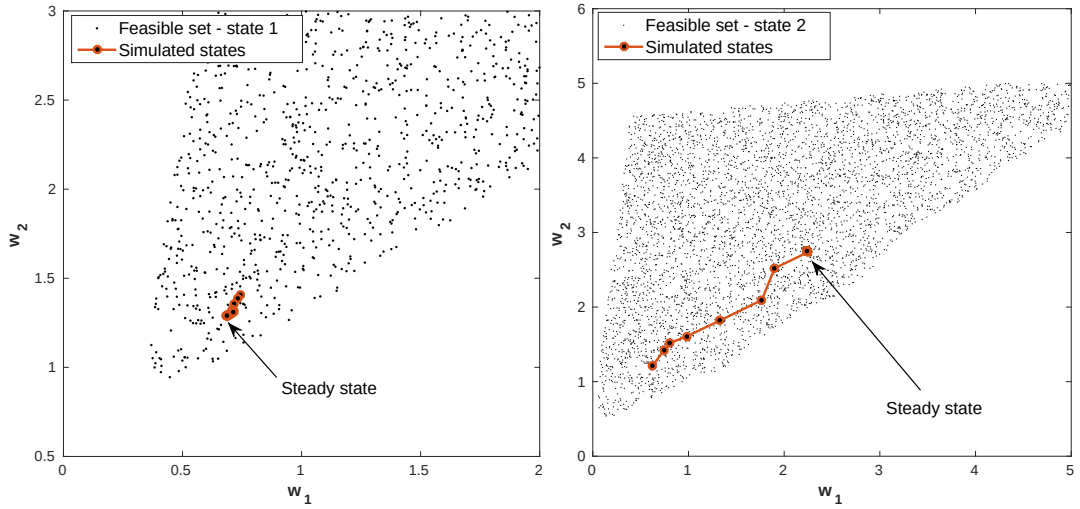

Figure 12: Left panel—simulation path in the feasible set for the low state, state 1. Right panel—the same, for the high state, state 2.

impulse–response experiments on the equilibrium policies. Starting from the steady state, we induce the complementary state, and then return to the original one. In Fig. 13, we can see that the perturbed quantities quickly return to the steady state both for the low and the high types. The reason for this is that the discount factor $\beta = 0.9$ is relatively small (see Tab. 1).

4.3 Many types

In this section, we demonstrate the dimensional scalability of our method. To do so, we expand the baseline model (see Sec. 4.1) to three types—a "low" (state 1), a "middle" (state 2), and a "high" (state 3) one. An agent's endowment is either $h_{low} = 0.3, h_{middle} = 0.6$, or $h_{high} = 0.9$. We set the discount factor to $\beta = 0.96$. Moreover, we choose the transition probabilities across the different types (see Theorem 3) as

$$\Pi_{\theta,\tilde{\theta}} = \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.2 & 0.5 & 0.3 \\ 0.0 & 0.4 & 0.6 \end{bmatrix}. \tag{44}$$

The choice of the Markov chain given by Eq. (44) is based on the following reasoning: Let us assume that an agent works in a firm with a rigid hierarchy. He can be promoted, demoted, or stay at his current rank. If the agent moves inside the company's hierarchy, he can do so only one level at a time. Moreover, it is also more likely that the agent is on a lower rung than it is for him to be of a high type. All other settings are kept as stated in Tab. 1.

The pre-computed equilibrium value correspondences for the three types, θ_1, θ_2, and θ_3, are depicted in the right panels of Figs. 14, 15, and 16, respectively. We can see that their three-dimensional geometry strongly resembles that of an airplane wing.

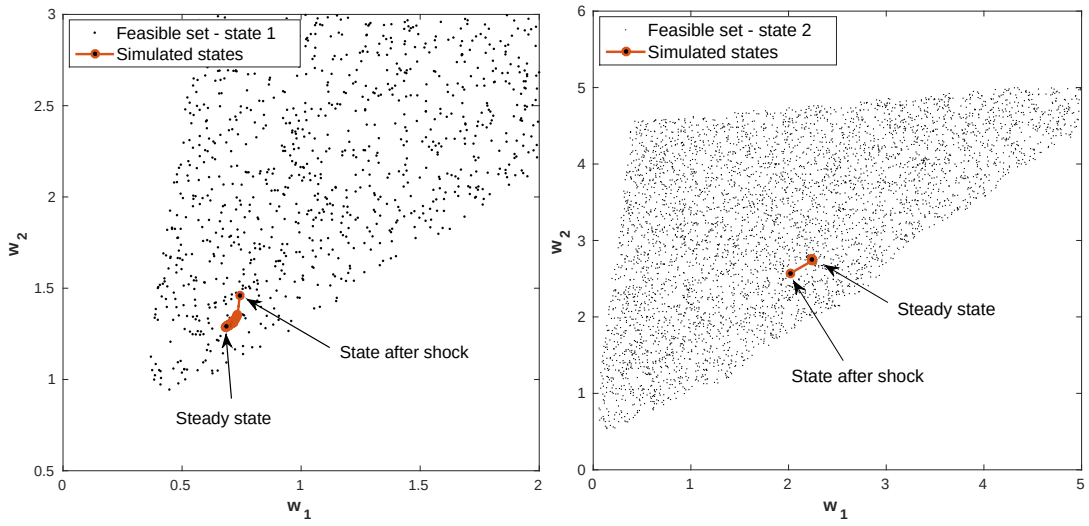

Figure 13: Path of the utility promises after a shock to the steady state has occurred, depicted in the left panel for state 1 and in the right panel for state 2. In both panels, the steady state as well as the utility promises right after the shock are labeled by arrows.

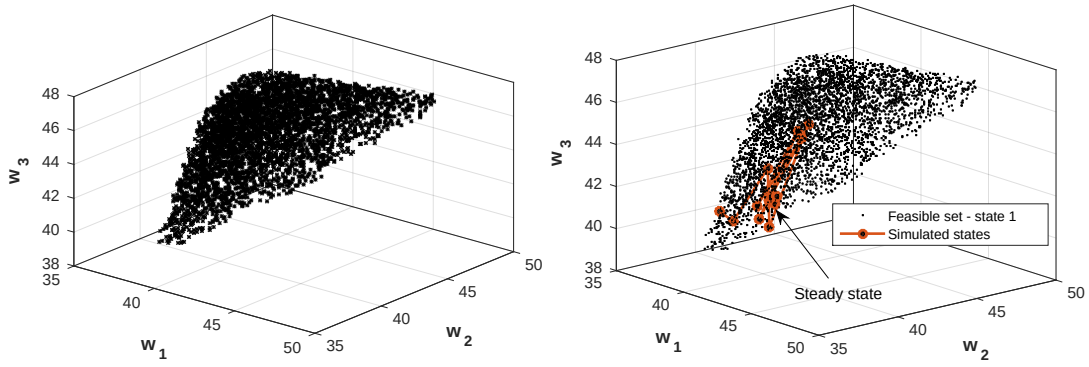

Figure 14: Left panel—the approximate feasible set $W(\theta_1)$ as a function of w_1, w_2, and w_3. Right panel—simulation path in the feasible set for the low state, state 1.

Next, we carry out a collection of simulations. The right panels of Figs. 14, 15, and 16 show the simulation paths within the three feasible sets. As in Secs. 4.1 and 4.2, we launched the simulations from the optimal value in the promised utility space and kept the shocks constant in the respective state.

In line with Sec. 4.1, we again find that in the low state, the optimal value and the steady state are close to one another and relatively low in the promised utility space (see Fig. 14). In contrast, in both the middle and high state, we start out with a low utility promise and then gradually raise it over time. Thus, we traverse a large fraction of the feasible set (see Figs. 15 and 16)—the same pattern that we found in the high state of the model with two types (see Sec. 4.1). The reason for such a simulation path is again that in the present model setting, there is no incentive constraint for the low type and only one for the middle type that prevents underreporting.

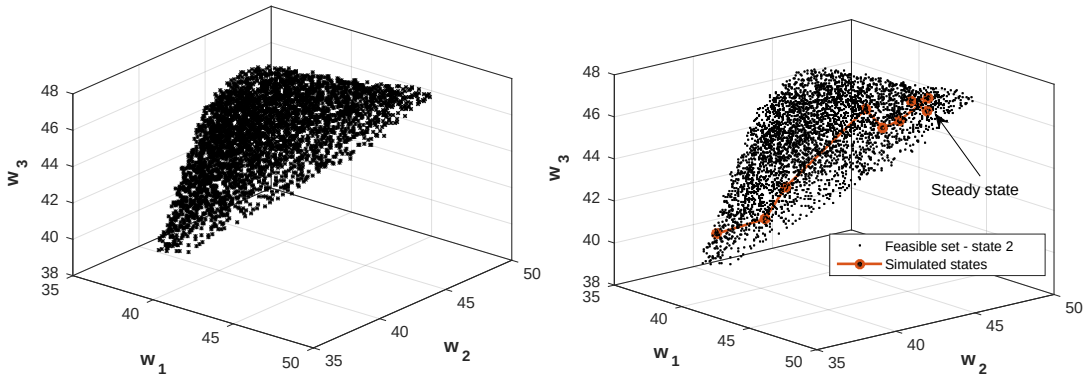

Figure 15: Left panel—the approximate feasible set $W(\theta_2)$ as a function of w_1, w_2, and w_3. Right panel—simulation path in the feasible set for the middle state, state 2.

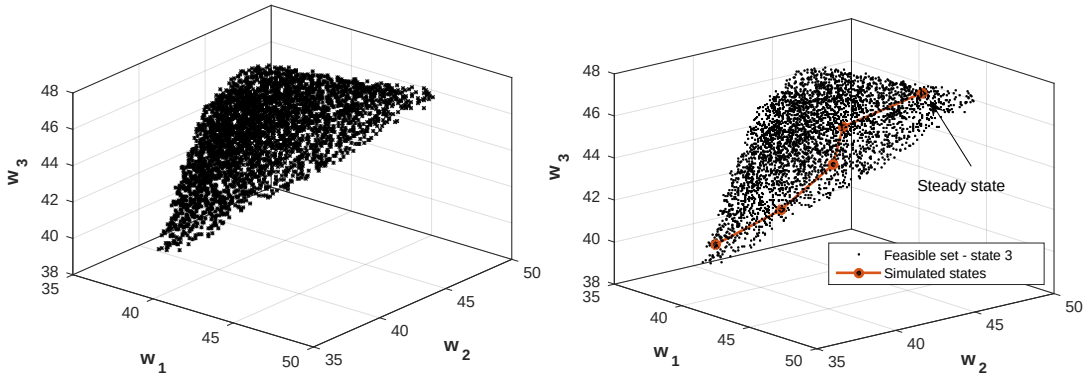

Figure 16: Left panel—the approximate feasible set $W(\theta_3)$ as a function of w_1, w_2, and w_3. Right panel—simulation path in the feasible set for the high state, state 3.

5 Conclusion

In this paper, we introduce a scalable and flexible framework for solving infinite-horizon, discrete time DI problems with hidden states and of unprecedented complexity. Addressing such dynamic adverse selection models is a formidable task, since two major computational bottlenecks create difficulties in the solution process. Issue one is related to the fact that the feasible sets of utility promises are not known in advance and have to be determined numerically. In most of the interesting economic applications, such sets have a non-hypercubic geometry and might even be non-convex. Issue two results from solving the dynamic adverse selection problem with VFI. To do so, we need to repeatedly approximate and evaluate high-dimensional functions at arbitrary coordinates within the domain of interest—that is, the feasible sets.

We address the computation of the equilibrium value correspondences, combining the set-valued dynamic programming techniques by APS with BGMMs. For the VFI, we use a massively parallel algorithm that applies GP machine learning in every iteration step to approximate the value functions on the pre-computed, time-invariant feasible sets.

Our approach provides several desirable features that yield a substantial improvement over the previous literature. First, since sampling achieves the approximation of feasible sets through BGMMs, it does not directly suffer from the curse of dimensionality and thus can deal with problems involving many types of agents. Second, our scheme of approximating equilibrium value correspondences is independent of solving the DI problem, and thus does not directly add to the computational complexity of solving the actual model. Third, it can approximate both convex and non-convex sets. Moreover, the equilibrium BGMMs can be used to generate sample data from within the approximate equilibrium value correspondences. This directly plays to the strength of GPR: a form of supervised machine learning that can—given a training set that is generated from within the feasible region—be used to approximate and interpolate functions on irregularly shaped state spaces. Furthermore, since the construction of GP surrogates is achieved by sampling from a domain of interest such as a feasible set, they can handle the curse of dimensionality to some extent. Thus, our proposed method—that is, using APS and BGMMs in conjunction with GPs—has the potential of handling highly complex DI models at a relatively low computational cost, as it focuses the resources where needed.

To demonstrate the capabilities of our framework, we compute solutions for models of repeated agency with history dependence, varying preferences, and increasingly many types. It is clear that while the focus of the work presented lies in solving dynamic adverse selection problems, the method proposed here has a far broader scope: it can also be applied, for example, to moral hazard problems, mechanism design, or to dynamic games, where one of the significant difficulties also lies in finding the equilibrium sets.

This all suggests that our framework will enable researchers to think of DI problems of greater richness than was possible prior to this work, as they no longer need to substantially restrict their modeling choices from the outset.

A Approximating feasible sets with adaptive sparse grids

To verify the functionality of the framework proposed in this paper for determining the equilibrium correspondences (see Sec. 3.2), we cross-validate it by applying adaptive sparse grids (see, e.g., Brumm and Scheidegger (2017)). In particular, we solve—as briefly mentioned in Sec. 3.2—an auxiliary problem of Eq. (20) for which we know the true solution on the feasible set. More precisely, we relax the feasibility via a penalty function such that the resulting value function will be zero on the feasible set and less otherwise. Strictly speaking, we are looking for the fixed point of the following dynamic program:

$$
\begin{aligned}
F(w,\theta) = \max_{c,w^+,\xi} \sqrt{\varepsilon} - \sqrt{\varepsilon + \sum_{\tilde{\theta}} \xi_{\tilde{\theta}}^2} + \beta \sum_{\tilde{\theta}\in\Theta} \Pi_{\theta,\tilde{\theta}} F^+(w_{\tilde{\theta}}^+, \tilde{\theta}) \qquad (45)\\
\text{s.t. } w_\nu = \xi_\nu + \sum_{\tilde{\theta}\in\Theta} \Pi_{\nu,\tilde{\theta}} (u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w_{\tilde{\theta},\tilde{\theta}}^+) \ \forall \nu \in \Theta,\\
u(c_{\tilde{\theta}}, \tilde{\theta}) + \beta w_{\tilde{\theta},\tilde{\theta}}^+ \geq u(c_\nu, \tilde{\theta}) + \beta w_{\nu,\tilde{\theta}}^+ \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta \setminus \{\tilde{\theta}\},\\
c \in [0, \overline{c}]^N,\\
w_{\tilde{\theta}}^+ \in \mathbb{R}^N \text{ with } w_{\tilde{\theta},\nu}^+ = (w_{\tilde{\theta}}^+)_\nu \ \forall \tilde{\theta} \in \Theta, \ \forall \nu \in \Theta,
\end{aligned}
$$

where $\varepsilon > 0$ is a relaxation parameter in

$$
\sqrt{\varepsilon} - \sqrt{\varepsilon + \sum_{\tilde{\theta}} \xi_{\tilde{\theta}}^2}, \qquad (46)
$$

and where Eq. (46) is a smooth approximation of the norm $\| \cdot \|_1$. $F(\cdot,\cdot)$ is approximated by piecewise linear basis functions on an adaptive sparse grid. Eq. (45) is an ordinary DP problem with a fixed point and therefore can be solved by VFI. We know that for all feasible w, the value function will be 0. To see this, note that the payoff of Eq. (45) in some iteration t is $\sqrt{\varepsilon} - \sqrt{\varepsilon + \sum_{\tilde{\theta}} \xi_{\tilde{\theta}}^2}$—that is to say, less than or equal to zero. Moreover, any feasible point will have an optimal value of zero, since there we will not need to relax the bounds on the equality constraints. Thus, the resulting value function has to be zero for feasible values and less than zero otherwise (for more details, see Judd et al. (2016)). Once the VFI for Eq. (45) has converged, we can use $F(\cdot,\cdot)$ as a penalty for the dynamic incentive problem (see Eq. (24)).

We now recompute the test case outlined in Sec. 4.1. In Fig. 17, we display the equilibrium correspondences for the low and high states, computed by applying adaptive sparse grids. Comparing Fig. 17 to Fig. 3, it becomes apparent that the two methods yield similar results and thus confirm each other.

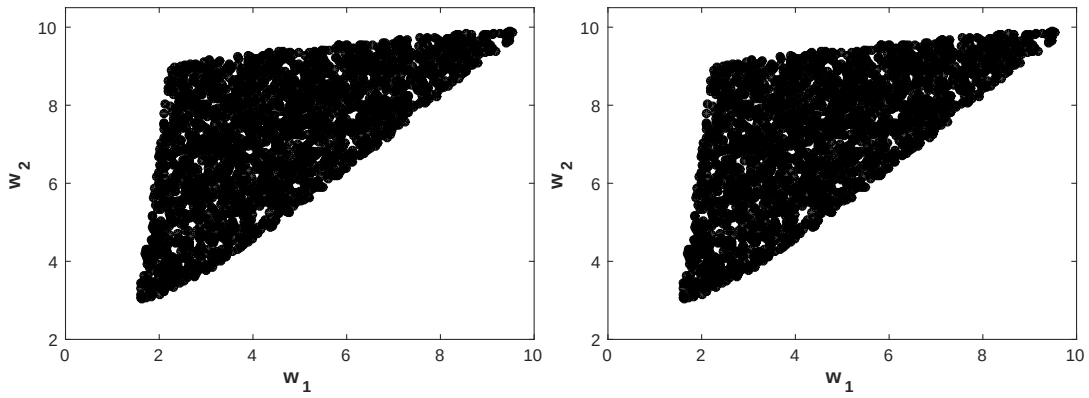

Figure 17: The left panel shows the approximate equilibrium correspondences for the low state as a function of w_1 and w_2, whereas the right panel shows the respective set for the high state. Both feasible sets were computed by applying adaptive sparse grids and set-valued DP (see Eq. (45)).

References

Abraham, Arpad and Nicola Pavoni. "Efficient allocations with moral hazard and hidden borrowing and lending: a recursive formulation". *Review of Economic Dynamics* 11.4 (2008), pp. 781–803.

Abreu, Dilip, David Pearce, and Ennio Stacchetti. "Optimal cartel equilibria with imperfect monitoring". *Journal of Economic Theory* 39.1 (1986), pp. 251–269.

Abreu, Dilip, David Pearce, and Ennio Stacchetti. "Toward a Theory of Discounted Repeated Games with Imperfect Monitoring". *Econometrica* 58.5 (1990), pp. 1041–1063.

Abreu, Dilip and Yuliy Sannikov. "An algorithm for two-player repeated games with perfect monitoring". *Theoretical Economics* 9.2 (2014), pp. 313–338.

Bellman, R. *Adaptive Control Processes: A Guided Tour*. 'Rand Corporation. Research studies. Princeton University Press, 1961.

Bertsekas, Dimitri P. *Dynamic Programming and Optimal Control*. 2nd. Athena Scientific, 2000.

Blei, David M. and Michael I. Jordan. "Variational inference for dirichlet process mixtures". *Bayesian Analysis* 1 (2005), pp. 121–144.

Broer, Tobias, Marek Kapicka, and Paul Klein. "Consumption risk sharing with private information and limited enforcement". *Review of Economic Dynamics* 23 (2017), pp. 170–190.

Brumm, Johannes, Dmitry Mikushin, Simon Scheidegger, and Olaf Schenk. "Scalable high-dimensional dynamic stochastic economic modeling". *Journal of Computational Science* 11 (2015), pp. 12–25.

Brumm, Johannes and Simon Scheidegger. "Using adaptive sparse grids to solve high-dimensional dynamic models". *Econometrica* 85.5 (2017), pp. 1575–1612.

Cole, Harold L. and Narayana R. Kocherlakota. "Efficient allocations with hidden income and hidden storage". *The Review of Economic Studies* 68.3 (2001), pp. 523–542. eprint: `/oup/backfile/content_public/journal/restud/68/3/10.1111/1467-937x.00179/2/68-3-523.pdf`.

DeMarzo, Peter M. and Yuliy Sannikov. "Optimal security design and dynamic capital structure in a continuous-time agency model". *The Journal of Finance* 61.6 (2006), pp. 2681–2724.

Doepke, Matthias and Robert M. Townsend. "Dynamic mechanism design with hidden income and hidden actions". *Journal of Economic Theory* 126.1 (2006), pp. 235–285.

Eftekhari, Aryan, Simon Scheidegger, and Olaf Schenk. "Parallelized dimensional decomposition for large-scale dynamic stochastic economic models". *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '17. Lugano, Switzerland: ACM, 2017, 9:1–9:11.

Fernandes, Ana and Christopher Phelan. "A recursive formulation for repeated agency with history dependence". *Journal of Economic Theory* 91.2 (2000), pp. 223–247.

Golosov, Mikhail, Aleh Tsyvinski, and Nicolas Werquin. "Recursive contracts and endogenously incomplete markets". *Handbook of Macroeconomics* 2 (2016), pp. 725–841.

He, Zhiguo, Bin Wei, Jianfeng Yu, and Feng Gao. "Optimal long-term contracting with learning". *The Review of Financial Studies* 30.6 (2017), pp. 2006–2065.

Judd, Kenneth L. "Projection methods for solving aggregate growth models". *Journal of Economic Theory* 58.2 (1992), pp. 410–452.

Judd, Kenneth L. *Numerical methods in economics*. The MIT press, 1998.

Judd, Kenneth L, Lilia Maliar, Serguei Maliar, and Rafael Valero. "Smolyak method for solving dynamic economic models: lagrange interpolation, anisotropic grid and adaptive domain". *Journal of Economic Dynamics and Control* 44 (2014), pp. 92–123.

Judd, Kenneth L., Philipp Renner, Simon Scheidegger, and Sevin Yeltekin. *Optimal Dynamic Fiscal Policy with Endogenous Debt Limits*. Working Paper. 2016.

Judd, Kenneth, Sevin Yeltekin, and James Conklin. "Computing supergame equilibria". *Econometrica* 71.4 (2003), pp. 1239–1254.

Krueger, Dirk and Felix Kubler. "Computing equilibrium in OLG models with stochastic production". *Journal of Economic Dynamics and Control* 28.7 (2004), pp. 1411–1436.

Lambert, Richard A. "Long-term contracts and moral hazard". *Bell Journal of Economics* 14.2 (1983), pp. 441–452.

Ljungqvist, L. and T.J. Sargent. *Recursive macroeconomic theory*. Mit Press, 2000.

Malin, Benjamin, Dirk Krueger, and Felix Kuebler. "Solving the multi-country real business cycle model using a smolyak-collocation method". *Journal of Economic Dynamics and Control* 35.2 (2010), pp. 229–239.

Meghir, Costas and Luigi Pistaferri. "Income variance dynamics and heterogeneity". *Econometrica* 72.1 (2004), pp. 1–32.

Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

Pavoni, Nicola, Christopher Sleet, and Matthias Messner. "The dual approach to recursive optimization: theory and examples". *Forthcoming, Econometrica* (2017).

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. New York, NY, USA: Cambridge University Press, 2007.

Rasmussen, Carl Edward. "The infinite gaussian mixture model". *In Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 554–560.

Rasmussen, Carl Edward and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Rogerson, William P. "Repeated moral hazard". *Econometrica* 53.1 (1985), pp. 69–76.

Sannikov, Yuliy. "A continuous- time version of the principal: agent problem". *The Review of Economic Studies* 75.3 (2008), pp. 957–984.

Scheidegger, Simon and Ilias Bilionis. "Machine learning for high-dimensional dynamic stochastic economies". *Preprint available at SSRN: https://ssrn.com/abstract=2927400* (2017).

Sleet, Christopher and Sevin Yeltekin. "On the computation of value correspondences for dynamic games". *Dynamic Games and Applications* 6.2 (1, 2016), pp. 174–186.

Spear, Stephen E. and Sanjay Srivastava. "On repeated moral hazard with discounting". *The Review of Economic Studies* 54.4 (1987), pp. 599–617.

Stokey, Nancy, Robert Lucas, Jr., and Edward Prescott. *Recursive Methods in Economic Dynamics*. Cambridge, MA: Harvard University Press, 1989.

Storesletten, Kjetil, Christopher Telmer, and Amir Yaron. "Consumption and risk sharing over the life cycle". *Journal of Monetary Economics* 51.3 (2004), pp. 609–633.

Waechter, Andreas and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Math. Program.* 106.1 (2006), pp. 25–57.

Werning, Ivan. *Optimal unemployment insurance with unobservable savings*. 2002.

Williams, Noah. "On dynamic principal-agent problems in continuous time" (2009).

Williams, Noah. "Persistent private information". *Econometrica* 79.4 (2011), pp. 1233–1275.

Yeltekin, Sevin, Yongyang Cai, and Kenneth L. Judd. "Computing equilibria of dynamic games". *Operations Research* 65.2 (2017), pp. 337–356.