

ESSAYS IN ECONOMETRICS AND DYNAMIC KIDNEY EXCHANGE

by

Vitor Baisi Hadad

Submitted to the Faculty of the Department of Economics
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy

Boston College
Morrissey College of Arts and Sciences
Graduate school

ProQuest Number: 10792678

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10792678

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

©Copyright 2018 Vitor Baisi Hadad

Essays in Econometrics and Dynamic Kidney Exchange

Vitor Baisi Hadad

Abstract This dissertation is divided into two parts.

Part I - Dynamic Kidney Exchange In recent years, kidney paired donation (KPD) has emerged as an attractive alternative for end-stage renal disease patients with incompatible living donors. However, we argue that the matching algorithm currently used by organ clearinghouses is inefficient, in the sense that a larger number of patients may be reached if kidney transplant centers take into consideration how their pool of patients and donors will evolve over time. In our work *Two Novel Algorithms for Dynamic Kidney Exchange*, we explore this claim and propose new computational algorithms to increase the cardinality of matchings in a discrete-time dynamic kidney exchange model with Poisson entries and Geometric deaths.

Our algorithms are classified into *direct prediction* methods and *multi-armed bandit* methods. In the *direct prediction method*, we use machine learning estimator to produce a probability that each patient-donor pair should be matched today, as opposed to being left for a future matching. The estimators are trained on offline optimal solutions. In contrast, in *multi-armed bandit* methods, we use simulations to evaluate the desirability of different matchings. Since the amount of different matchings is enormous, multi-armed bandits (MAB) are employed to decrease order to decrease the computational burden.

Our methods are evaluated using simulations in a variety of simulation configurations. We find that the performance of at least one of our methods, based on multi-armed bandit algorithms, is able to uniformly dominate the myopic method that is used by kidney transplants in practice.

We restrict our experiments to pairwise kidney exchange, but the methods described here are easily extensible, computational constraints permitting.

Part II - Econometrics In our econometric paper *Heterogenous Production Functions, Panel Data, and Productivity*, we present methods for identification of moments and nonparametric marginal distributions of endogenous random coefficient models in fixed- T linear panel data models.

Our identification strategy is constructive, immediately leading to relatively simple estimators that can be shown to be consistent and asymptotically normal. Because our strategy makes use of special properties of “small” (measure-zero) subpopulations, our estimators are *irregularly* identified: they can be shown to be consistent and asymptotically Normal, but converge at rates slower than root- n .

We provide an illustration of our methods by estimating first and second moments of random Cobb-Douglas coefficients in production functions, using Indian plant-level microdata.

Table of Contents

I Dynamic kidney exchange 1

Two novel algorithms for dynamic kidney exchange 2

1	Introduction	2
1.1	Related literature	4
2	Background and definitions	9
2.1	Environments	11
2.2	ABO Environment	11
2.3	RSU Environment	12
2.4	OPTN Environment	13
3	Methods	15
3.1	Objective and benchmarks	15
4	Algorithms	18
4.1	Direct prediction	18
4.2	Multi-armed bandit methods	20
5	Results	28
5.1	Direct prediction methods	28
5.2	Multi-armed bandit methods	29
6	Extensions and future work	33
7	Conclusion	35
8	Appendix	36

II Econometrics 50

Heterogenous Production Functions, Panel Data, and Productivity

Dispersion		51
1	Introduction	51
2	Background and literature review	53
3	Model and assumptions	58
4	Identifying random coefficient moments	60
4.1	First period moments	60
4.2	Second period moments	63

4.3	Identifying further moments	64
4.4	Extension: fixed coefficients	65
4.5	Identifying the marginal distribution of random coefficients	65
5	Estimation	67
6	Asymptotic results	70
7	Simulations	71
7.1	Bootstrap coverage	73
8	Empirical application	74
9	Conclusion and future work	83
10	Appendix	84
10.1	Summary statistics	84
10.2	Simulation results	85
10.3	Python module FHHPS	90
	Bibliography	93

List of Figures

1	Patient and donor cPRA	15
2	Comparing MYOPIC and OPT	17
3	Direct prediction method	18
4	Multi-armed bandit methods (one-period)	25
5	Function GET_PSEUDO_REWARD	26
6	Intuition for pseudo-rewards	27
7	Performance of gradient boosting in direct prediction method	29
8	Performance of logistic regression in direct prediction method	30
9	Performance of random forests in direct prediction method	30
10	Performance of multi-armed bandit method in ABO and RSU environments	31
11	Performance of multi-armed bandit method in OPTN environment	32
12	Linear panel data models	57
13	Shock estimates in simulated data	78
14	Random coefficient estimates in simulated data	79
15	Shock estimates using ASI data	80
16	Random coefficient estimates using ASI data (2008)	81
17	Random coefficient estimates using ASI data (2009)	82

List of Tables

1	Performance of statistical methods as classifiers	37
2	Performance of direct prediction method with gradient boosting (1)	38
3	Performance of direct prediction method with gradient boosting (2)	39
4	Performance of direct prediction method with logistic regression (1)	40
5	Performance of direct prediction method with logistic regression (2)	41
6	Performance of direct prediction method with random forests (1)	42
7	Performance of direct prediction method with random forests (2)	43
8	Performance of multi-armed bandit method using UCB1	44
9	Performance of multi-armed bandit method using Thompson sampling	45
10	Performance of multi-armed bandit method in ABO and RSU environments	46
11	Performance of multi-armed bandit method in OPTN environment	47
12	Performance of multi-armed bandit method in OPTN environment	48
13	Approximate comparison to a <i>chunk-matching</i> algorithm	49
14	RMSE-minimizing parameter configurations for our DGP.	73
15	Bootstrap coverage for shock moments	74
16	Bootstrap coverage for random coefficient conditional moments	74
17	Bootstrap coverage for random coefficient unconditional moments	74
18	Empirical application: Shock Estimates	77
19	Empirical application: Random Coefficient Estimates (first period)	77
20	Empirical application: Random Coefficient Estimates (second period)	77
21	ASI survey data: summary statistics	84
22	ASI survey data: correlations	84
23	Transformed variables: summary statistics	85
24	Transformed variables: correlations	85
25	Performance measures for estimates of $E[U_2]$	86
26	Performance measures for estimates of $E[V_2]$	86
27	Performance measures for estimates of $\text{Std}[U_2]$	86
28	Performance measures for estimates of $\text{Std}[B_2]$	87
29	Performance measures for estimates of $\text{Cov}[U_2, V_2]$	87
30	Performance measures for estimates of $\text{Corr}[U_2, V_2]$	87
31	Performance measures for estimates of $E[A_1]$	88

32	Performance measures for estimates of $E[B_1]$	88
33	Performance measures for estimates of $Std[A_1]$	88
34	Performance measures for estimates of $Std[B_1]$	89
35	Performance measures for estimates of $Cov[A_1, B_1]$	89
36	Performance measures for estimates of $Corr[A_1, B_1]$	89
37	Linear panel regressions	91
38	Quantile panel regressions	92

Part I

Dynamic kidney exchange

Two novel algorithms for dynamic kidney exchange

VITOR HADAD

1 Introduction

Patients suffering from end-stage renal disease have two available treatments: dialysis and renal transplant. While transplant is associated with “lower mortality rates and improved quality of life compared to chronic dialysis treatment” (Tonelli et al., 2011), severe kidney shortages prevent tens of thousands of patients every year from receiving a transplant. In the US alone, there are currently over 90,000 patients in the kidney transplant waiting list. These patient may wait several years before finding an available donor, and many will die – often of co-morbidities – before receiving a transplant.

In order to overcome this challenge, *kidney paired donation* (KPD) has emerged as an alternative option for patients who have an incompatible but otherwise willing living donor. This approach, conceived about three decades ago by Rapaport (1986), involves pooling patient-donor pairs and swapping or exchanging donors so as to increase the overall number of matched patients. Nowadays, KPD programs have become common both in the United States and internationally, and patients are actively encourage to participate in them. (LaPointe Rudow et al., 2015)

Nevertheless, the total amount of kidney exchanges still remains smaller than its potential. As we will argue in this paper, one source of inefficiency in KPD is that most transplant centers do not take into consideration the evolution of the pool of donors and pairs over time, which can lead to large long-run losses in terms of the overall cardinality of matched pairs.

To overcome this issue, in this paper we contribute to the *dynamic* kidney matching literature by introducing two novel algorithms for kidney exchange. They are explained next.

Main idea This paper proposes two novel approaches for increasing the cardinality of matched pairs in a discrete-time dynamic model of kidney exchange: the *direction prediction* and the *multi-armed-bandit* methods.

First, our *direct prediction* method recasts the dynamic kidney exchange problem as a binary classification task: for each pair in the pool, we predict a binary label representing whether they should be matched today (one), or left for the future (zero), given their observable characteristics. This prediction is produced by an estimator trained in a large a number of simulations that were solved by an offline solver.

The main challenge with this approach is how to represent the data that will be fed to the estimator. While covariates such as ABO blood type and current waiting times can be naturally represented by real numbers, it is not immediately clear how to represent the relationship between pairs in a kidney matching pool, which is a global, graph-theoretic concept. We attempt to overcome this issue by augmenting the data set with graph-theoretic notions such as measures of node centrality, size of graph, average degree, and so on.

Our second method, named the *multi-armed bandit* (MAB) method, uses simulations to answer the question: “if we commit to matching one particular set of pairs today, how likely are we to decrease the *overall* number of matched pairs between

now and a horizon h ?”. This probability can be estimated via simulations. However, the main challenge here is that simulations are so computationally costly, and the number of available actions so vast, that we are cannot realistically produce accurate estimates of it for every element of the action space. To overcome this issue, we employ multi-armed bandit algorithms. Their role is to manage a relatively small computational budget and determine which actions are more attractive (and should be explored further), and which are not (so not worth exploring).

Main results Our methods are evaluated entirely via simulations. The simulation setups, which we call *environments*, are inspired by models used the kidney exchange literature, and differ by their rules regarding blood- and tissue-type compatibility.

We compare our method against two benchmarks: an algorithm that clears the maximal matching at each period, called MYOPIC; an infeasible offline optimal algorithm that we call OPT. The former roughly represents what most organ clearing-houses are doing today, while the latter represents the maximal payoff we could have achieved, had we been able to know the future. Our metric for comparison is the *per-period average number of matched pairs* over a long period of time.

In all environments, we observe that our implementation of the *direct estimation* method fails to consistently improve upon MYOPIC. However, the *multi-armed bandit* (MAB) method is able to improve uniformly and substantially upon the MYOPIC benchmark. In particular, we observe that our MAB does particularly well in environments of moderate sparsity.

1.1 Related literature

This paper pulls ideas from microeconomics, matching theory, and operations research, and uses methods drawn from computer science and machine learning. Let us take a brief look at these fields in turn.

1.1.1 Kidney Exchange

Early years In microeconomics, the literature on kidney exchange began with Roth et al. (2004), who provided a kidney exchange mechanism inspired by the *housing* problem studied in the literature of mechanism design as in Shapley and Scarf (1974) and later Abdulkadiroğlu and Sönmez (1999). While their mechanism had significant theoretical advantages, it also relied on large number of exchanges being conducted simultaneously, and drew criticism for making assumptions about heterogeneous preferences over kidneys. Subsequent work by Roth et al. (2005) addressed some of these criticisms by focusing on logistically simple mechanisms that used only 2-way exchanges, and assumed also that patients were indifferent between all compatible kidneys. In a later paper, Roth et al. (2007) demonstrated via simulations that allowing for 3-way exchanges in addition to 2-way exchanges could increase the cardinality of matched pairs by a great deal, but larger cycles would only bring about modest improvements.

Recent advances in static exchange In the last ten years, both in academia and in medical practice the focus began to shift from exchanges via cycles to exchanges emanating from *non-directed donors* (NDD) – altruistic donors who are willing to donate their organs to anyone. Such exchanges yield a *chain* of transplants that begins with the NDD and may either terminate with the last pair donating to a waitlist recipient in a kidney registry, or not terminate at all and have the last pair’s donor await an opportunity to become a future living donor. In the former kind of exchange, called *domino paired donation* by Montgomery et al. (2006), all transplants occur simultaneously. In the second kind, transplants may be spread over several months. For this reason, the latter kind of transplant is called *non-simultaneous, extended, altruistic donor chain* Rees et al. (2009).¹

¹On occasion, NEAD chains have also been called *Never-ending altruistic donor chains*, as relayed by (Roth, 2015, p. 235-6).

A significant amount of attention has been devoted to these NEAD chains, including notably Ashlagi et al. (2011) and Ashlagi et al. (2012), whose results show that NEAD chains benefit highly sensitized patients in sparse pools of moderate size, because they decrease the need for simultaneous double-coincidence of wants in the exchange market. In fact, Anderson et al. (2015) reports that “NEAD chains are responsible for the majority of successful kidney transplants conducted via kidney exchange at both the [Allied for Paired Donation] and within other major exchange programs”.

Dynamic kidney exchange The literature concerning the *dynamic kidney exchange* problem, where we take into account the evolution of the kidney exchange pool over time, is much smaller. It began with a seminal paper by Ünver (2010), who derives a dynamic mechanism that produces optimal n-way cyclic exchanges in the steady state of a continuous-time model with Poisson arrivals. The results in that paper rely on three simplifying assumptions. First, that the waiting cost is constant for all pairs; second, that pairs do not leave the pool unless they are matched; finally, that pairs within the pool are only blood-type incompatible. It can be shown that, under these assumptions, all pairs are rendered homogeneous. This dramatically decreases the dimension of the state space, and allows for an easily interpretable solution.

Ashlagi et al. (2013) note that the graphs of real-life pools tend to be sparse and filled with pairs that are either easy or extremely hard to match. In order to model this particular feature, they postulate a sparse heterogeneous random graph model containing only two such types, and propose a greedy algorithm that waits until the pool contains a certain number of pairs of each type, and then matches as many as pairs as possible. In a different vein, Akbarpour et al. (2017) studies the relationship between “market thickness” (the number of available pairs in the exchange pool) and matching time in a simple model of stochastic arrival and departure. They show

that, in their model, greedy algorithms that cleverly exploit the time to match can perform close to infeasibly optimal benchmarks, even if these algorithms are ignorant about the global structure of the graph, and have no information about agents' departure times.

In the computer science and operations research literature, Abraham et al. (2007) and more recently Anderson et al. (2015), Dickerson et al. (2016) and Dickerson et al. (2017) have mostly focused on producing scalable combinatorial programming algorithms that can take on static matching problems with large graphs and allowing for both chains and cycles of moderately large length. However, a series of papers starting with Awasthi and Sandholm (2009) and followed by Dickerson et al. (2012) and Dickerson and Sandholm (2015) have dealt with the dynamic kidney problem by *weighted myopia*. Their idea is to prevent wasteful matchings (e.g., an O-donor to an AB-patient) by artificially introducing negative weights to graph components containing them. In Dickerson and Sandholm (2015), these optimal weights are computed from simulations involving historical data.

Computer science and machine learning We were inspired by several computational alternatives to dynamic kidney exchange that have been proposed in the past few years. In particular, the *multi-armed bandit method* described here is reminiscent Algorithm 1 in Awasthi and Sandholm (2009), inasmuch it also selects the best actions today by repeatedly simulating the future, solving an offline problem, computing a “score” for each cycle, and selecting actions with maximum score. However, both our simulation and “scoring” methods differ importantly because we leverage multi-armed bandit algorithms to decrease the computational burden of producing multiple simulations.

Dickerson et al. (2012) and Dickerson and Sandholm (2015) propose a related method that they call *weighted myopia*. The idea is to use simulations for learning

“potentials” of specific elements of the graph. Much like the “scores” in Awasthi and Sandholm (2009), the set of “potentials” associated with a specific exchange encodes how desirable each exchange is in terms of its future value. The authors then use these pre-learned numbers to revise the weights in a myopic algorithm, forcing it to select more desirable exchanges where it would be otherwise indifferent. Our *direct prediction method* works similarly, with three crucial differences. First, we do not directly predict how “desirable” a matching will be, but instead we predict whether or not an offline, optimal algorithm would choose the node or not. Second, our method uses a more aggressive thresholding mechanism to select which nodes should be matched today, and which should be left for later. Third, we optionally make use of information about how the node fits inside the compatibility graph, so that the information about the node (e.g., the blood type and HLA profile of patient and donor) is augmented with graph-theoretic notions (e.g., measures of node centrality, degree, etc).

Sequential decision problems Zooming out of our application, our problem lies within a larger class of sequential decision problems whose central feature is a concern with the *exploration-exploitation trade-off*. That is, problems that can be described as the one faced by an agent who must spend a limited computational budget to *explore* a certain action space so as to find and *exploit* actions that will maximize her expected rewards. Crucially, the agent sequentially observes rewards for actions that she has taken, but does not observe rewards for other actions. This class also encompasses, for example, *Markov decision process* (MDP) and its variants.

A *multi-armed bandit*² (MAB) problem is a particular version of an MDP where time is discrete, and at every period an agent chooses from a finite and fixed number

²The terminology comes from a turn-of-the-20th-century United States colloquialism, when slot machines were called “one-armed bandits”. While the name apparently suggests a loss in the long run, the “bandit” problems studied in academic literature do not necessarily have negative expected payoff.

of actions K that produce stochastic rewards. The objective is usually to minimize the amount of *regret* that accumulates over time, where *regret* is loosely defined as the difference between the agent's actual accumulated rewards under her own strategy and an the average reward that the agent would have gotten had she chosen the optimal action at each period since the beginning. (We will make this definition more rigorous in a later section). Crucially, at every period the agent only observer rewards for actions that she has chosen, and not for other actions.

Multi-armed bandit problems were studied sporadically in the last century, with the earliest reference going as far back in time as Thompson (1933). Interest was rekindled after a seminal paper by Lai and Robbins (1985), who developed a strategy that provably attains an asymptotic lower bound of regret. Later, their analysis was simplified by Agrawal (1995) who also developed a finite-time analysis of regret. In recent years, other alternative algorithms have been proposed. In particular, in our paper we use the *Thompson sampling* algorithm studied by Agrawal and Goyal (2012) and Kaufmann et al. (2012), and the *upper-confidence bound* (UCB1) algorithm developed by Agrawal (1995) and Auer et al. (2002). For an approachable review of multi-armed bandits, optimal strategies, and their variations, we refer the reader to the recent book by Lattimore and Szepesvari (2018).

2 Background and definitions

Medical background We say that a patient and a donor are *compatible* if the donor's organ cells do not present antigens that are capable of inducing an aggressive response by the patient's immune system. A successful transplant usually requires patients to be *blood-type compatible* and *tissue-type compatible* (also known as *histo-compatible*).

Blood-type compatibility refers to compatibility with respect to major ABO blood

groups. For example, *O*-type patients can only receive from *O*-type donors, *AB* patients may receive from any blood group, etc).

For the purposes of this paper, a patient is tissue-type will be called *compatible* with a donor if the donor does not present alleles of gene complex called the *human leukocyte antigens* (HLA) that are deemed unacceptable by the patient's immune system.

Technical definitions We model the dynamic kidney exchange problem in a manner similar to Ünver (2010) and Akbarpour et al. (2017).

A *kidney exchange pool* is a directed random graph process $G_t = (V_t, E_t, X_t), t \in \mathbb{N}$ whose vertices $v \in V_t$ represent (patient, donor)-pairs³. A directed edge $e \in E_t$ between vertex v and v' means that the donor in pair v can donate to the patient in pair v' . When such an edge exists, we say that v is *compatible with* v' .

Each pair is endowed with certain characteristics $x \in X_t$ ⁴ such as patient and donor blood type (other characteristics will be discussed below).

We say that a pair *enters* the kidney exchange pool when it first becomes available for exchange. After a certain number of periods, the pair *leaves* the pool once and for all, or *dies*. The difference between entry and death is called a pair's *sojourn*.

An *exchange* is an ordered tuple of nodes $m = (v_{i_1}, \dots, v_{i_k})$ where each pair in the tuple is compatible with the next pair in the sequence. A *matching* is a set of exchanges where no pair appears in more than one exchange.

An *environment* is, informally speaking, a collection of rules governing which pairs are deemed compatible, the entry and death processes that govern the evolution of the pool, and which vertex characteristics are observable in and relevant to the problem. More formally, it can be defined as the conditional probability distribution between two kidney exchange pools, given the current pool G_t and G'_t , given a matching M_t .

³Throughout, we will use the terms *pair*, *node* and *vertex* interchangeably.

⁴In an slight abuse of notation, we will also denote by X_t the matrix of pairs' observable characteristics.

We will expand on this below. Finally, a *dynamic matching algorithm* is a procedure that selects matchings M_t at every period. Once a matching M_t is selected, all associated vertices and their edges are removed from the kidney exchange pool.

2.1 Environments

Our paper presents three environments inspired by previous works on dynamic kidney exchange. All three have the following assumptions in common.

First, the number of new incoming pairs in each period is drawn from the $Poisson(r)$ distribution, where $r \in \mathbb{N}$ denotes the *entry rate*, and also equals the expected number of entrants per period. Second, each pair independently draws the length of their sojourn from the $Geometric(d)$ distribution. The parameter $d \in \mathbb{R}$ is the *death rate*, and its reciprocal $\frac{1}{d}$ is the expected sojourn length. We note that, due to the memoryless property⁵ of the Geometric distribution, the amount of time a pair has waited in the pool gives us no information about how much time they have until their death. Third, we assume that patients do not discriminate between compatible kidneys. This last assumption translates as patients having binary preferences over kidneys (i.e., they receive utility 1 if they receive a transplant, and 0 otherwise), and is consistent with other works in the literature, notably Roth et al. (2005).

In what follows we will explain the differences between each environment.

2.2 ABO Environment

In the *ABO environment*, compatibility between two distinct pairs is based only on blood-type compatibility.

Blood types are drawn independently for patients and donors from the corresponding probability distribution in the US population⁶, but are adjusted by the following

⁵If $X \sim Geometric(p)$, then $P(X > t + s | X > s) = P(X > t)$

⁶Roughly O:49%, A:36%, B:11%, AB:4%

assumption previously used in Ünver (2010): we allow for incompatibility between a donor and their own patient. The reason for this additional assumption is that, if there were truly no tissue type compatibilities, we would never observe pairs of type (AB, \cdot) , (\cdot, O) , or (A, A) , (O, O) , (B, B) , and (AB, AB) ⁷, since their donors would be automatically compatible with their patients and they would never participate in an exchange. Following Zenios et al. (2001), we assume that for such patients the probability that a donor and their patient are incompatible is $p_c = 0.11$. Arrival rates are then adjusted accordingly, e.g., the arrival rate of (A, O) pair is proportional to $0.48 \times 0.36 \times 0.11 \approx 0.019$.

2.3 RSU Environment

The *RSU* environment is named after a classic simulation model in used Roth et al. (2007) and Saidman et al. (2006). Each pair is characterized by patient and donor ABO blood types, current waiting time, and a *calculated panel reactive antibody (cPRA)* level that represents the probability of a crossmatch with a random donor.⁸ The lower the cPRA, the higher the number of potentially compatible pairs.

The simulation process is as follows. First, we draw a pair in the same manner as in the ABO environment. Next, we draw if the patient is a female (with probability around 41%), and if so we also draw whether her donor is her husband (spouses comprise about 49% of donors). Finally, we draw a cPRA level for the patient (Low: 70.1%, Medium: 20%, High: 9.9%). This cPRA level determines the probability that they can receive a kidney from any donor, including their own: patients with low cPRA have a 5% probability of positive crossmatch with a random donor; patients

⁷For shorthand, we will sometimes write “An (X, Z) pair” to mean “any pairs where the patient has blood type X , and the donor has blood type Z ”.

⁸The original Roth et al. (2007) paper called this simply *PRA*, and in real life there is an important distinction between the two measures. However, for the purposes of a simulation model this distinction is immaterial. We keep the name *cPRA* for consistency with OPTN environment later.

with medium cPRA have a 45% chance, and patients with high cPRA have a 90% chance of a crossmatch. If a patient is blood or tissue-type incompatible with their own donor, they enter the pool. In addition, if the patient is female and her husband is the donor, the probability of positive crossmatch for low, medium and high cPRA patients goes up to 28.75%, 58.75% and 92.25%. This last adjustment reflects the fact that women tend to produce antibodies against their husbands' antigens during pregnancy.

Once in the pool, the pair immediately forms directed edges with the existing pairs, again following the patient cPRA distribution. The resulting random graph is akin to a Erdős-Rényi $G(n, p)$ random graph where the probability of forming edges is heterogeneous across different pair types.

2.4 OPTN Environment

In the *OPTN environment*, we use historical data collected by the United Network for Organ Sharing (UNOS) data provided in the Standard Research and Analysis (STAR) dataset. The STAR dataset contains information from all patients that were ever registered to the kidney waiting list in the United States for the past three decades, as well as from all living donors that actually participated in an transplant⁹. From this original dataset, we excluded entries associated with the following:

- Patients that were registered for more than one organ (including those who were simultaneously waiting for kidney and pancreas)
- Patients that were not waiting for their first kidney transplant
- Donors and patients with incomplete tissue-type profile information.

⁹To our knowledge, there is no centralized dataset containing information about registered donors that never went to transplant.

The resulting dataset contained 117813 patients and 9337 living donors. We call this the “historical dataset”.

Patient and donor cPRA We added two additional variables to the original dataset: a *patient cPRA* and a *donor cPRA*. While the patient cPRA is a measure of patient tissue-type incompatibility with a random donor Cecka (2010), our donor cPRA is a measure of the opposite direction – how frequently a donor is tissue-type incompatible with a random patient.¹⁰ Both were computed empirically: for each patient our dataset, we checked the how many donors exhibited antigens that are unacceptable for the patient in any of the A, B, Bw, C, DR, DPB, DQ, and DQA loci, and assigned this positive crossmatch probability as their patient cPRA¹¹; for the donors, we worked in the opposite direction by calculating the frequency of patients who exhibited antibodies against their donor’s antigens, and that became their donor cPRA. Figure 1 shows the distribution across the entire population.¹²

Artificial dataset We created an artificial dataset by randomly drawing patients and living donors from the historical dataset and checking for blood-type compatibility, and tissue-type compatibility as explained above. Compatible pairs were discarded. We iterated in this manner to construct a dataset of about one million incompatible pairs. At every simulation period, a random number of pairs is drawn from this dataset.

This artificial dataset is intended to approximate realistic kidney pools faced by actual clearinghouses. However, let us remark that our donor pool consists only of

¹⁰We thank Itai Ashlagi for the suggestion of a donor cPRA.

¹¹A genetic *locus* is a specific position on a chromosome. The loci above encodes a donor tissue type.

¹²We should remark that we found a very different patient cPRA distribution than the one in the OPTN dataset. This may have been because: in real life cPRA is computed using deceased donor data, while we used living donor data; we may have used different HLA equivalence tables; OPTN uses a more sophisticated model based on population genetics. Organ Procurement and Transplantation Network (2013)

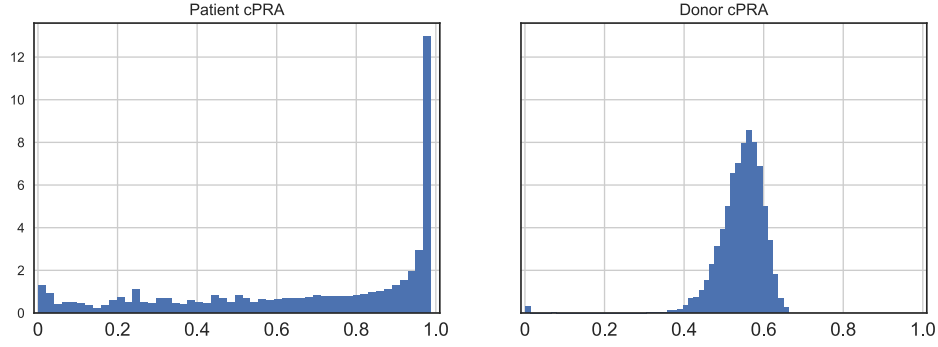


Figure 1: Patient and donor cPRA

Computed from historical data. Our patient cPRA is the frequency of living donors that exhibit antigens that are unacceptable to the patient. We also define a donor cPRA by calculating the frequency of patients that have antibodies against the donors antigens.

persons that actually underwent a transplant. Therefore, our dataset’s donors may be easier to match than a random donor from the entire pool of registered donors. Even then, a large fraction of the patients is highly sensitized, which mirrors what we see in KPD pools Ashlagi et al. (2013).

3 Methods

3.1 Objective and benchmarks

Objective In this work, we focus on maximizing the undiscounted cardinality of matched pairs over T periods. To make comparison to our benchmarks easier, we formulate the problem equivalently as maximizing the per-period average matching size over T periods.

Benchmarks Let \mathcal{M} a set of available matchings at period t .

$$\max_{M \subset \mathcal{M}} \sum_{m \in M} w_m x_m \quad (1)$$

$$\text{s.t.} \quad \sum_{m: v \in m} x_m \leq 1 \quad \forall v \in V \quad (2)$$

$$x_m \in \{0, 1\} \quad \forall m \in \mathcal{M} \quad (3)$$

where \mathcal{M} is the set of available matchings¹³, w_m is the cardinality of the exchange m , and x_m is a binary variable indicating whether or not the pair was selected. Constraint 3 ensures that each vertex is selected only once.

By an *static matching problem at period t* , we mean a version the problem above where the only available matchings involve pairs $v \in V_t$. By an *offline matching problem between t and s* , we mean a version of the same problem where involving vertices $v \in \cup_{k=0}^s V_{t+k}$, and any matching $m = (v_1, \dots, v_k)$ has the property that the sojourns of the donating pair overlaps with the sojourn of the receiving pair.

We are interested in how our new methods perform relative to the following two benchmarks.

Myopic At every period, the MYOPIC algorithm solves a static matching problem, finds the maximal matching and clears it immediately. In doing so, it disregards all observable characteristics of each pair, and in particular it ignores that some pairs might be useful to keep certain pairs may be easier or harder to match. Therefore, it may forgo the opportunity of matching a hard-to-match patient today, or postpone an easy-to-match pair for later. In essence, this is an approximation to what kidney exchanges currently do.

¹³By a small abuse of notation, we may write $v \in M$ to mean that v belongs to some exchange that is an element of the matching M

Optimal (OPT) This infeasible algorithm (henceforth OPT) solves the offline matching problem encompassing all periods between 1 and T . This completely does away with the uncertainty arising from the temporal structure of the problem, hence we know that this is the maximum achievable utility

Remark *How much is there to be improved upon?* In Figure 2, we compare MYOPIC and OPT for a grid of different entry and death rates. These results show that the performance gap between the two can be fairly large, in particular in sparser environments like *RSU* and *OPTN*. We also note that the gap is narrower when: the death rate is high, because if most pairs will die soon, dynamic considerations play a smaller role; and when the entry rate is very large, because in a thicker market pairs are able to encounter a suitable match more easily.

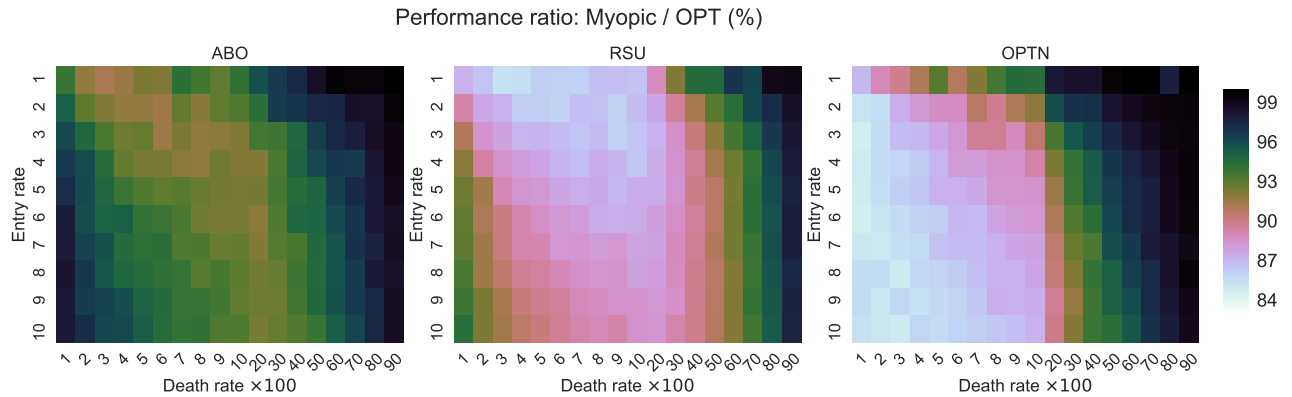


Figure 2: Comparing Myopic and OPT

Ratio of average per-period matched pairs for different entry and death rates, over 3000 periods (darker hues are better). Myopic has better chances of achieving performances similar to OPT when the death rate is high (moving rightwards on the graphs), or when entry rate is high (moving downwards on the graphs).

4 Algorithms

We now present two novel methods to determine which patients should be matched. Note that in this paper we will be working solely with pairwise matchings (i.e. matchings that only involve two-cycles), but the methods below are straightforwardly extensible to matchings involving larger cycles and chains.

4.1 Direct prediction

The main insight exploited by this method is that we can break down the dynamic kidney problem into two parts. The first is determining *which* pairs should be matched today, and the second is deciding *how* the selected pairs should be matched among themselves. Note also, that the second part of the problem can be solved immediately as an integer programming problem.

Using the idea above, the *direct prediction* method essentially reduces the dynamic matching problem to a classification task: at each period, we aim to produce a binary label for each node indicating whether it should be matched in this period (1) or left for later (0). Selected nodes are then passed to a static solver that finds the maximal matching among them. Once these nodes are cleared, time evolves to the next period.

The procedure is formalized in Algorithm 1, and also illustrated in Figure 3.

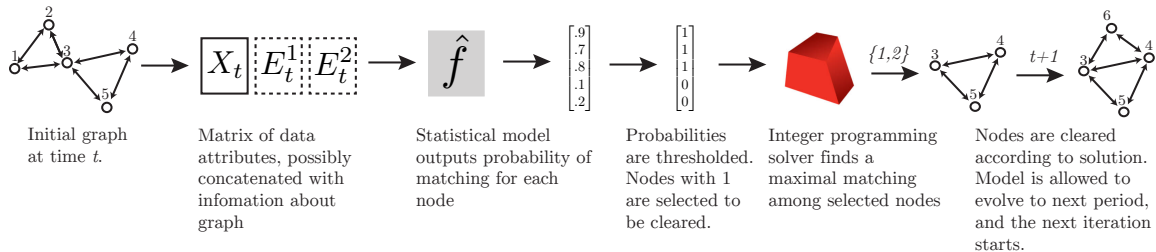


Figure 3: Direct prediction method

One period of simulation using direct prediction methods to choose cycles. See Algorithm 1 for details.

ALGORITHM 1: DIRECT_PREDICTION Method

Data: Environment simulator object ENV;
Integer programming solver object SOLVER;
Statistical method CLASSIFIER

Threshold *thres*;

Function DIRECT_PREDICTION(ENV, SOLVER, CLASSIFIER, *thres*):

```
// Retrieve node (and potentially also graph) data from current environment
X, E1, E2 ← ENV.GET_DATA()
// Classifier predicts matching probability for each pair using data
prob ← CLASSIFIER(X, E1, E2)
// Get index of pairs whose probability is higher than threshold
index ← WHICH(prob > thres)
// Find maximal matching restricted to this subset
chosen_cycles ← SOLVER.SOLVE(ENV, SUBSET=index)
// Return chosen cycles to be cleared
return chosen_cycles
```

Training the classifier In order to produce data to feed into our “classifier”, we repeatedly created 1000-period simulation runs and solved them offline using OPT. Then, for every period t in simulation k , we stored: a matrix X_t^k whose rows represent each pairs observable characteristics; an additional matrices E_t^{1k} containing additional graph-theoretical information such as several centrality measures (betweenness, in-degree, out-degree, harmonic, closeness), in- and out-degrees, and average neighbor degree; a conforming matrix E_t^{2k} containing the entry and death rates used in that simulation; a binary vector y_t^k indicating which nodes OPT chose to match in period t .

We repeated this procedure for each environment, and produced three artificial data sets, each containing approximately 2 million observations. These data sets were then fed to a series of predictive algorithms: penalized logistic regression (Wu et al., 2009), random forest classifiers (Breiman, 2001), and gradient tree boosting classifiers (Friedman, 2001).

4.2 Multi-armed bandit methods

Since we are assuming access to the exact data-generating process, we can use simulations to choose the best action at each period t : choose a certain action (such as removing a specific cycle); simulate the evolution of the kidney exchange pool until some time horizon $t + h$; solve the offline problem between t and $t + h$; evaluate the performance of the chosen action under some criterion; repeat until we have evaluated the performance of all actions; finally, take the best-performing action.

However, the approach outlined above often turns out to be impossible to execute in practice, because simulations are computationally expensive and in practice we cannot repeat them enough times get reliable estimates of the average performance for each action choice, especially in large graphs. In order to solve this problem, we will leverage theory and some algorithms from the multi-armed bandit (MAB) literature described next.

In addition, we also need to specify exactly what is the criterion under which each the performance of each action will be measured. In order to do that we will specify a measure based on a new concept that we call *pseudo-rewards*.

Our procedure is illustrated in Figures 4 and 5.

At the beginning of the period t , the agent receives a set of cycles C that are available to be cleared.¹⁴ If C is empty, nothing happens and we move to the next period. Otherwise, the agent then picks a cycle $c \in C$ and simulates the future, including new entries and deaths, up to a horizon h . Next, OPT is run twice, once normally, and once with the additional constraint that c be removed today. The size of the resulting matching in these two scenarios is compared. Naturally, the constrained version of OPT cannot achieve anything better than its unconstrained counterpart, but it might get to be equal. If it is, the agent receives a *pseudo-reward* of one, otherwise

¹⁴Here we talk about cycles for simplicity, and because our application only uses two-cycles. In the general case, the agent may receive an arbitrary set \mathcal{M} of available exchanges.

it receives zero.

This process is repeated: at each iteration ℓ , a cycle c_ℓ is chosen and its pseudo-reward $r_{c,\ell}$ is revealed. When a preset computational budget of $L(|C|)$ iterations is hit, the agent then analyses the whole history of cycle choices and pseudo-rewards $H_t = \{(c_\ell, r_{c\ell})\}_{\ell=1}^L$, and decides whether to match one of the cycles or move on to the next period. If a cycle c is chosen it is immediately cleared, however the environment does not evolve to the next period yet. Instead, the history H_t is discarded the procedure is repeated again with a reduced set of actions $C' \subset C$ that produces a new history H'_t and so on, until either there are no more available choices or the agent decides to allow the environment to move on to time $t + 1$. When that at last happens, entries and deaths are revealed, the agent receives a new set of cycles, and the process begins anew. All past information is ignored.

An important detail was left out of the explanation above: how does the agent chooses the next cycle to test and simulate at each iteration ℓ ? To answer that, we will take a detour to give a self-contained introduction about multi-armed bandit algorithms.

Interlude: Multi-armed Bandit Algorithms Let c^* be a cycle that maximizes expected pseudo-rewards during one round of the algorithm:

$$c_\ell^* \in \arg \max_c E[r_{c,\ell}]$$

Also, let *regret* be defined as the difference between expected reward of the optimal choice c^* and its own selected choice c_ℓ .

$$\Delta_\ell := E[r_{c^*,\ell}] - E[r_{c,\ell}]$$

A *multi-armed bandit* (MAB) algorithm is an adaptive exploration procedure that

seeks to minimize the cumulative regret over L rounds. A good MAB algorithm will act so as to balance exploration (trying out different choices to get high-quality estimates of their rewards) and exploitation (using out better choices more often to increase total rewards), and produce regret that grows at an asymptotically slow rate. Here, we experiment with two common bandits algorithms, namely *UCB1*, and *Thompson sampling*. The literature on bandit algorithms is extensive and an in-depth explanation is outside the scope of this paper. However, for context in the next paragraphs we provide some intuition for how and why they work.

UCB1 The *Upper Confidence Bound 1* prescribes that at each period of repeat the following two steps.

1. Construct a certain confidence interval around the average reward estimate for each arm.
2. Choose the action with the highest confidence upper bound.

This heuristic is commonly named *optimism in the face of uncertainty* (Kaelbling et al., 1996), because at every period we are choosing the action with the “largest plausible” average reward estimate (Lattimore and Szepesvari, 2018, Ch. 3). The intuition is the following: if the agent is correct in choosing the optimistic action, then they receive zero regret; if the agent is incorrect, then they will thereafter review their estimates so as to decrease the upper bound for that action and avoid it in the future.

Auer et al. (2002) proved that, in the absence of further information about the distribution of rewards, by appropriately constructing the confidence interval around each reward average we are able to attain an optimal logarithmic cumulative regret

rate. Their proposed confidence interval for action c in round ℓ has the form

$$UCB[\ell, c] = \hat{\mu}_{c,\ell-1} + \kappa \sqrt{\frac{\log(\ell)}{n_{c,\ell-1}}} \quad (4)$$

where ℓ is the current round of the bandit algorithm, $\hat{\mu}_{c,\ell-1}$ is the running reward estimate for action c , and $n_{c,\ell-1}$ is the number of times that the action c has been selected before round ℓ , and κ is a constant whose optimal value is 2 when rewards are binary.

It is instructive to remark that Equation 4 is increasing in $\hat{\mu}_{c,\ell}$ – so that actions with higher expected reward will be naturally chosen more often, leading to exploitation – but it is also decreasing in $n_{c,\ell-1}$ – making other actions more likely to be chosen in the future, leading to more exploration.

Thompson sampling Also known as *posterior sampling*, this algorithm reformulates the bandit problem in a Bayesian framework. The agent begins with a prior distribution $P(\mu_c)$ for each action c . This initial prior is updated as the history of actions and rewards accumulates. Like UCB1, the entire algorithm is explained in few steps:

1. Compute and sample from the posterior distribution of average rewards

$$\tilde{\mu}_c \sim P(\mu_c | H_{\ell-1}) \quad \forall c$$

2. Choose the action with maximal reward among the samples.

$$c_\ell = \arg \max_c \mu_c$$

This simple algorithm can be shown to be optimal, in terms of attaining a asymptotically logarithmic lower bound on Bayesian regret (Slivkins, 2018). However, more

surprisingly, Kaufmann et al. (2012) showed that it also enjoys good frequentist properties, and under certain conditions performs no worse than UCB1 in terms of *frequentist* regret. Moreover, Thompson sampling algorithm can be generalized in a number of ways, some of which we discuss in section 6.

ALGORITHM 2: MULTI_ARMED_BANDIT Method

Data: Environment simulator object ENV;

Integer programming solver object SOLVER;

Threshold *thres*; Horizon *h*;

Function MULTI_ARMED_BANDIT(ENV, SOLVER, BANDIT, *thres*, *h*):

```

done ← False
while not at end of this document do
    // Select a cycle or a null token
    c ← CHOOSE_CYCLE(ENV,h,thres)
    // Check if a cycle was indeed chosen
    if c is not NULL then
        // Remove cycle and continue search
        ENV.REMOVE(c)
    else
        // Just terminate search
        done ← True
    end
// Return environment with removed cycles
return ENV

```

ALGORITHM 3: Function GET_PSEUDO_REWARD

Data: Cycle *c*; Horizon *h*;

Lists pseudo-reward statistics *Avg*, *Std*;

Environment simulator object ENV;

Oracle solver object OPT;

Function GET_PSEUDO_REWARD(ENV, *c*, *Avg*, *Std*, *h*):

```

// Simulate up to horizon h and find optimal matching
ENV.SIMULATE(h)
r1 ← OPT.SOLVE(ENV)
// Remove cycle c, find constrained optimal matching
ENV.REMOVE(c)
r2 ← OPT.SOLVE(ENV)
// Return 1 if rewards are equal, 0 otherwise
return r1 == r2

```

What are pseudo-rewards? As we match and clear out a cycle *c* today, we forgo the opportunity of using any future cycles involving the nodes in *c*. However, because

ALGORITHM 4: Function `choose_cycle`

Data: Horizon h ; Number of iterations L ; Threshold $thres$;

Environment simulator object `ENV`; Multi-armed bandit algorithm object `MAB`;

Function `choose_cycle(ENV, h, thres):`

```
// Initialize lists of current pseudo-reward statistics
C ← ENV.GET_AVAILABLE_CYCLES()
Avg ← ZEROS(LENGTH(C))
Std ← ZEROS(LENGTH(C))
// Begin iterations
for i ← 0 to L do
  // Bandit algorithm chooses next cycle to test given statistics
  c ← MAB.PULL(C, AVG, STD)
  // Compute pseudo reward for this cycle and update statistics
  r ← GET_PSEUDO_REWARD(ENV, c, Avg, Std, h)
  Avg ← UPDATE_RUNNING_AVERAGE(Avg, r)
  Std ← UPDATE_RUNNING_STD(Std, r)
end
// Bandit algorithm chooses best cycle given statistics
c.best ← MAB.CHOOSE(C, AVG, STD)
// Return best cycle, unless none of the pseudo-reward averages are above a certain threshold
if ALL(Avg ≤ thres) then
  | return NULL
else
  | return c
end
```

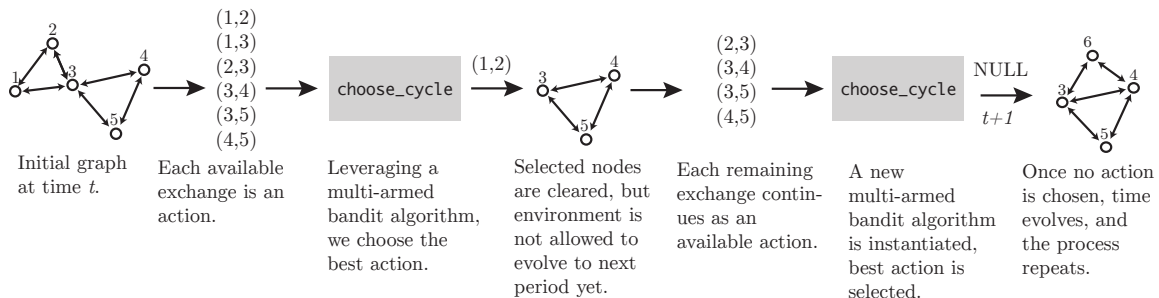


Figure 4: Multi-armed bandit methods

One period of simulation using multi-armed bandit methods to choose cycles.

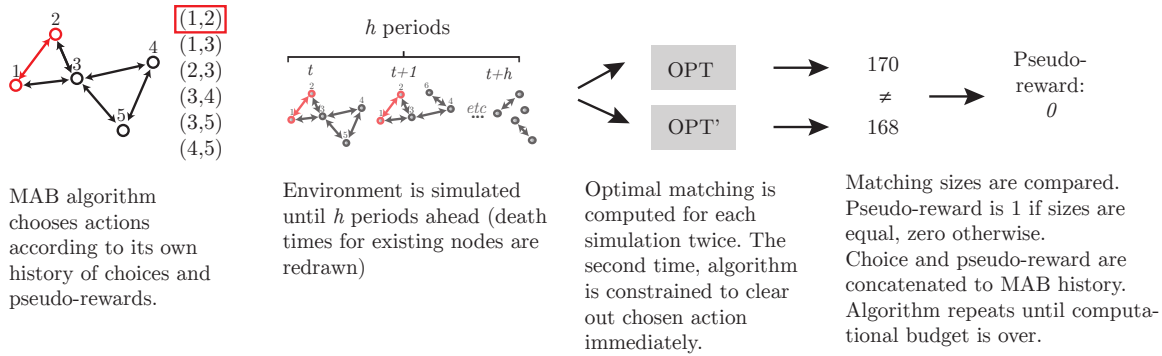


Figure 5: Function GET_PSEUDO_REWARD

Multi-armed bandits evaluate if a cycle should be cleared by checking if there is a high chance that the cycle will be used in the future. Such cycles get a *lower* reward, and are left for later.

there may be multiple optimal matchings, sometimes the cycles that become unavailable due to the removal of the nodes in c do not matter, in the sense that we can still find a matching of the same cardinality without them. In other words, we pay no price for removing these future cycles.

The pseudo-reward associated with cycle c is a random variable whose expectation is the probability that removing a cycle today will *not* negatively impact the optimal matching size between t and $t+h$. The higher this number, the more confident we are that clearing c out today will not give us trouble in the future. A concrete example of this idea is shown in Figure 6.

Pseudo-rewards are a natural way for us to control which patients should be matched today. Pairs that are easy to match will likely belong to many cycles, so by removing them we will be incurring a large cost in terms of future cycles that will become unavailable. But that means that the pseudo-reward associated with cycles that involve them will be lower, making them less attractive. On the other hand, patients that are harder to match will not participate many future exchange, so the price we pay for matching them today is low, and their average pseudo-reward is high.

Cycles at t		Cycles appearing between $t+1$ and $t+h$ (simulated)										Maximal matching sizes		Pseudo-reward						
												Unconstrained	Constrained							
$(1, 2)$	$(1, 3)$	$(2, 6)$	$(3, 5)$	$(3, 5)$	$(3, 6)$	$(4, 5)$	$(4, 5)$	$(4, 5)$	$(4, 6)$	$(4, 7)$	$(5, 6)$	$(5, 6)$	$(6, 7)$	$(6, 7)$	$(6, 8)$	$(6, 9)$	$(7, 9)$	$(8, 9)$	8/8	1
$(1, 2)$	$(1, 3)$	$(2, 6)$	$(3, 5)$	$(3, 5)$	$(3, 6)$	$(4, 5)$	$(4, 5)$	$(4, 5)$	$(4, 6)$	$(4, 7)$	$(5, 6)$	$(5, 6)$	$(6, 7)$	$(6, 7)$	$(6, 8)$	$(6, 9)$	$(7, 9)$	$(8, 9)$	10/8	0
$(1, 2)$	$(1, 3)$	$(2, 6)$	$(3, 5)$	$(3, 5)$	$(3, 6)$	$(4, 5)$	$(4, 5)$	$(4, 5)$	$(4, 6)$	$(4, 7)$	$(5, 6)$	$(5, 6)$	$(6, 7)$	$(6, 7)$	$(6, 8)$	$(6, 9)$	$(7, 9)$	$(8, 9)$	8/8	1
$(1, 2)$	$(1, 3)$	$(2, 6)$	$(3, 4)$	$(3, 4)$	$(3, 4)$	$(3, 5)$	$(3, 5)$	$(3, 5)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	$(4, 6)$	6/6	1

Figure 6: Intuition for pseudo-rewards

Each row in the example above, is supposed to represent one simulation. The three tuples on the left, $(1, 2)$, $(1, 3)$ and $(3, 4)$, represent the three cycles that are available to be matched today. The tuples to the right indicate cycles that appear once we simulate h periods into the future. In this example, we are contemplating selecting and clearing cycle $(1, 2)$ at t .

As explained in Figure 5, at each simulation we solve the problem twice: once where all cycles between t and $t + h$ are available, and a second time where we are constrained to match cycle $(1, 2)$ at t . In the constrained case, all other cycles that involve pair 1 or pair 2 become unavailable (in red).

In three out of our four simulations we were able to clear the same number of cycles in the constrained scenario, so the running pseudo-reward average is $\frac{3}{4}$. This indicates that, more often than not, removing the cycle $(1, 2)$ today will not decrease the overall long-run payoff, so matching it today is desirable.

5 Results

5.1 Direct prediction methods

Table 1 shows the performance results for direct prediction methods as estimators, i.e., how accurately they are able to predict whether a node should be matched or not. We see that while overall accuracy can be relatively high at 70-80%, precision (defined as the ratio between true positives and both true and false positives) is low even for simpler environments like *ABO*, indicating that the models are over-predicting matchings. Also, augmenting the data with information about the graph has little discernible effect under any of the performance criteria. This suggests that there might be gains from using other algorithms that make better use of information about the compatibility graph.

In order to empirically evaluate the performance of our proposed direct prediction method, we proceed as follow. First, we simulate a dynamic kidney pool according the configuration prescribed by each of our environments, and applied Algorithm 1 at each time step, following the outline on Figure 3. Next, using the *same* random seed, we simulated each environment and solved it again using MYOPIC in place of our method. Finally, we computed the average number of matched patients in 1000 periods for each algorithm, and compared the two.

This process was repeated several times for: each classifier (logistic regression, gradient tree boosting, random forests), each environment (*ABO*, *RSU*, *OPTN*) and nine entry and death rate combinations. The threshold variable was fixed at $thres = \frac{1}{2}$ everywhere.

The result is shown on Figures 7, 8 and 9, where we show the average performance ratio computed as above for each combination of environment, algorithm and entry/death rate configuration. (See also accompanying tables 2-7 in the Appendix).

Unfortunately, as we predicted in the beginning of the section, the poor predictive

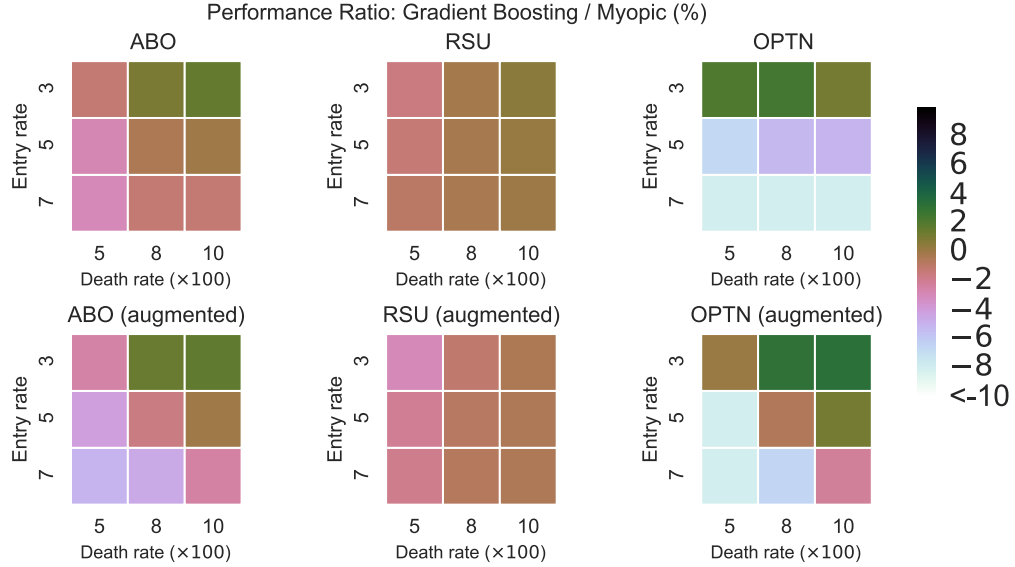


Figure 7: Performance of gradient boosting in direct prediction method
Ratio of average matched patients against MYOPIC. Simulations ran for 1000 periods.
Lower row is when data was augmented with information about graph.

performance of the algorithms translates into poor performance as the classifier in our direct prediction method. Random forests, in particular, have the lowest rate of accuracy (Table 1), and also exhibit the lowest performance in the direct prediction method.

5.2 Multi-armed bandit methods

We evaluated the performance of our *multi-armed bandit* in an analogous manner to the direct prediction method described in the previous section.

The average ratio between our method and MYOPIC is shown on Figures 10 and 11 (See also Tables 9-11). The results suggest that, at least for the entry and death rate combinations we experimented on, the *multi-armed bandit* method uniformly dominates MYOPIC in terms of average number of matched patients per period.

In sparser environments (*RSU* and *OPTN*), for particular entry and death rate combinations, our method is able to improve upon MYOPIC, sometimes over 4%. This reflects an earlier observation we did when analyzing Figure 2: gains from taking dy-

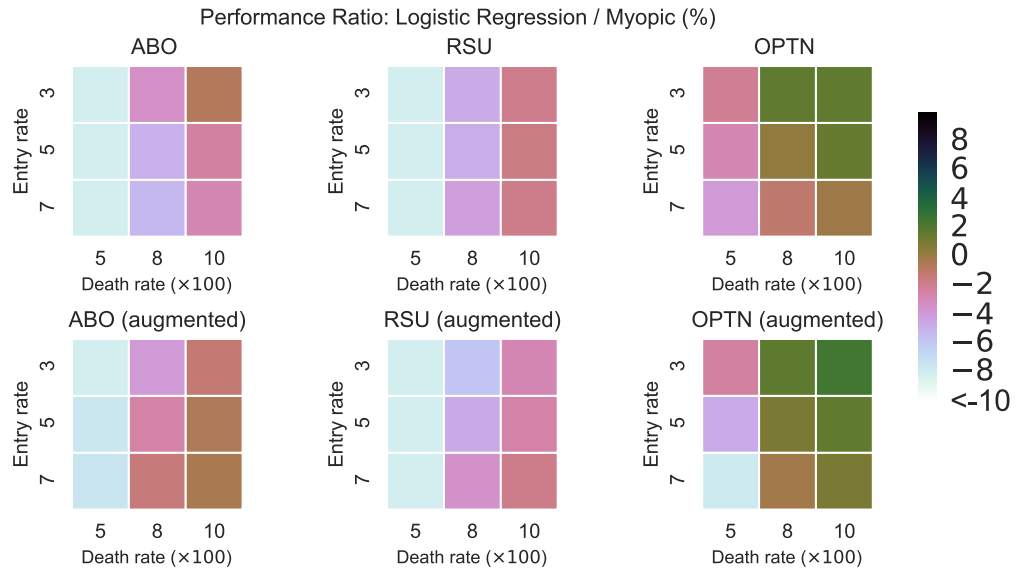


Figure 8: Performance of logistic regression in direct prediction method
 Similar to gradient boosting shown in Figure 7, logistic regression is only able to perform better than MYOPIC when the death rate is large.

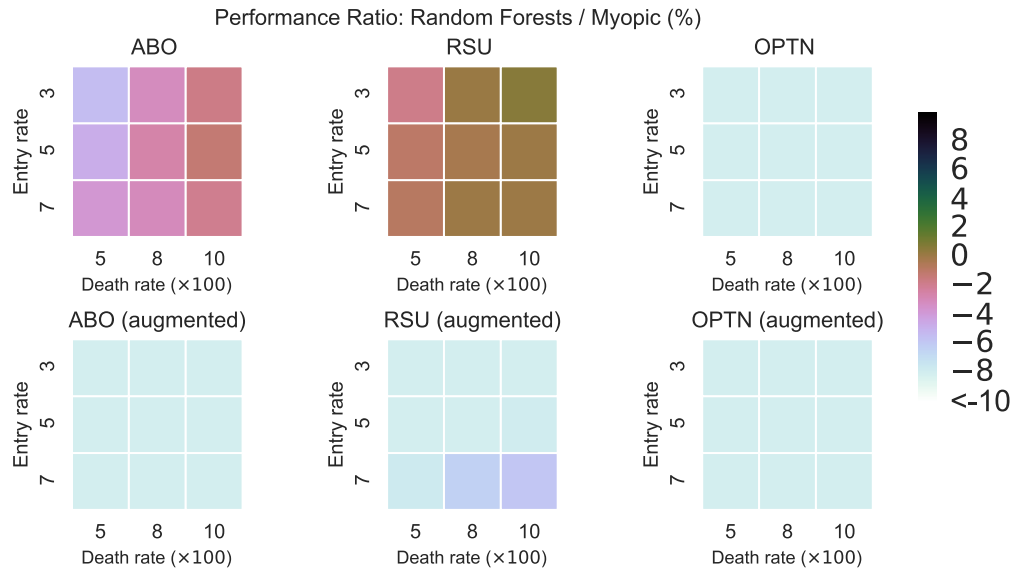


Figure 9: Performance of random forests in direct prediction method
 Random Forests's poor performance is likely due to its low accuracy, as shown on Table 1.

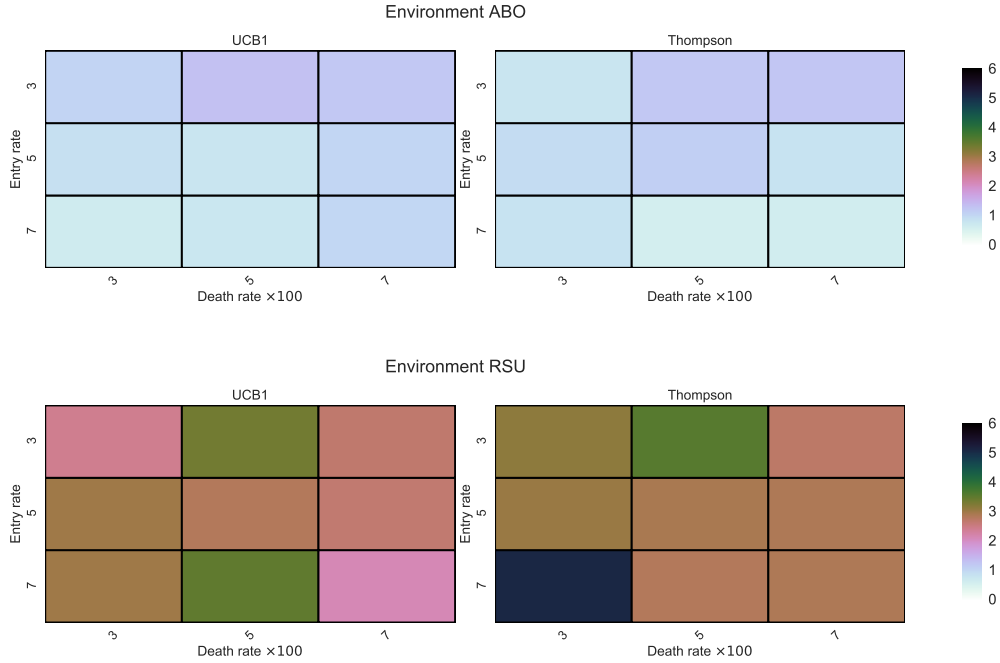


Figure 10: Performance of multi-armed bandit method in ABO and RSU environments

Average percent improvement by the *multi-armed bandit* method over MYOPIC. See also Tables 9 and 10.

dynamic consideration into account are larger in situations of moderate sparsity, because that is where MYOPIC forces the pool to be too thin, and drives the performance away from optimality.

Aside: Comparison with *chunk* algorithms Recently there has been some interest algorithms that apply the MYOPIC algorithm (or some appropriate variation of it) at every k number of periods. A notable example of this is Ashlagi et al. (2013), who propose the following dynamic kidney model: at each period, only one pair enters the pool; this pair is of one of two types; there are no deaths. For this model, derive a greedy algorithm called *Chunk Matching* (CM), which finds and clear the maximal matching each time a given number of each type has joined the pool.

We cannot compare Ashlagi et al. (2013)'s method directly to ours, since our models are slightly different. However, we could still consider the following thought

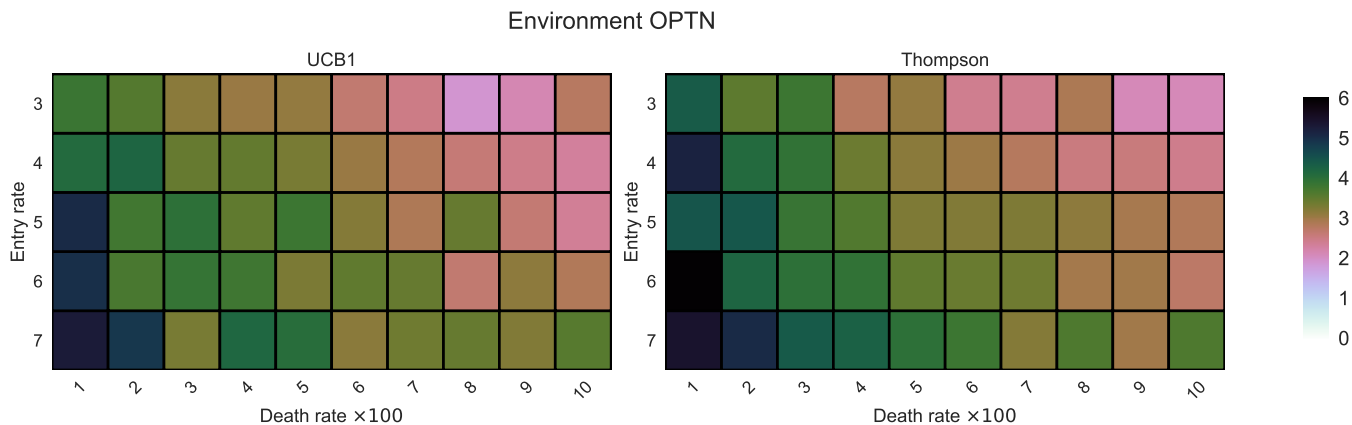


Figure 11: Performance of multi-armed bandit method in OPTN environment
 Average percent improvement by the *multi-armed bandit* method over MYOPIC.
 See also Table 11.

experiment: if we allowed only one pair per period, but ran MYOPIC every λ periods, would we see an increase in gains?

It is possible to give an approximate answer to this thought experiment using the results presented in this section. The reason is this. In our original model, λ pairs enter per period and each pair has a probability $1 - d$ of surviving to the next period. This is roughly equivalent to a model where we subdivide the period into λ sub-period, allow one pair to enter per period, and each pair has a probability $(1 - d)^{\frac{1}{\lambda}}$ of surviving to the next sub-period.¹⁵ Moreover, running MYOPIC at every period in the original model is approximately the same as waiting for λ periods in the modified model with deflated death rates.

We perform this comparison for the OPTN model, and present them on table 13. As we readily sees, our algorithm would still dominate the chunk-MYOPIC algorithm described above.

¹⁵This is exact for pairs that are already in the pool at time t , since they will survive with probability $((1 - d)^{\frac{1}{\lambda}})^{\lambda} = 1 - d$. However, a pair that arrives, e.g. in the second sub-period between t and $t + 1$ will reach $t + 1$ with slightly higher probability, since it must survive for fewer periods. We will ignore this in our rough approximation.

6 Extensions and future work

Our methods suggest a variety of extensions.

Improving the direct prediction method First, the relatively weak performance of our *direct prediction* method could be improved in two ways. First, by improving our ability to use information about graph structure when predicting if a node should be matched or not. This could be done using new statistical methods that are applicable to non-euclidean spaces (Shuman et al., 2013). In particular, recent years have seen the emergence of interest in generalizing neural networks to model with structured datasets such as graphs. A promising approach is the one taken by Kipf and Welling (2016), who build on earlier work by Defferrard et al. (2016) to derive a very simple recurrence relation for graph embedding:

$$H^{\ell+1} = \sigma((A_t + I)H^\ell W^\ell) \quad \text{with} \quad H^0 = X_t \quad (5)$$

where X_t is the matrix of observable characteristics, W^ℓ is a matrix of coefficients, A_t is the adjacency matrix associated with the kidney exchange pool at t , I is a conformable identity matrix, and σ is a nonlinear function such as the inverse logistic CDF or ReLU¹⁶. The authors prove that this can be interpreted as a differentiable version of the Weisfeiler-Lehman algorithm for graph isomorphism tests. For us, it is an especially convenient method because the input graphs need not be the same size throughout the analysis.

Second, when the *direct prediction* method computes a selection probability for each pair, it does not consider correlations between multiple pairs in the same kidney exchange pool. However, intuition suggests that it should be often the case that pairs in the same exchange have similar probabilities of being matched. In order to rectify

¹⁶ $ReLU(x) = \max(x, 0)$

this shortcoming of our approach, we would like to train our statistical estimator using the sequence-to-sequence (*seq2seq*) models of recurrent neural networks studied by Sutskever et al. (2014), Kalchbrenner and Blunsom (2013) and Cho et al. (2014). Sequence-to-sequence models are a relatively new paradigm that allows for variable length inputs (needed here because different pools have different sizes), and correlated outcomes of variable length. (See also (LeCun et al., 2015) for an overview)

Improving the MAB algorithm Our *multi-armed bandit* method can be quite computationally intensive, requiring multiple simulations per exchange. When the exchange pool or the set of available actions is large, it may be difficult to scale. It is important to consider extensions to deal with this issue.

First, note how our MAB algorithm only makes use of simulations to select exchange that will be clear, and it makes no use explicit use of information about pairs characteristics, or about the structure of the graph. However, it is intuitive that pairs endowed with similar characteristics should have similar probabilities of being matched. In terms of the model discussed in this paper, this would translate into correlated pseudo-rewards among pairs, where the strength of the correlation should be proportional to how similar two pairs are deemed to be. The generalized versions of the Thompson Sampling algorithm (See Russo et al. (2017)) are able to deal with this, and could be employed.

Other extensions We would like to experiment with different objective functions, since, in reality, different exchanges may have different levels of desirability or priority. For example, it is common for pediatric patients and previous organ donors receive higher priority, as do *ABDR0* exchanges involving “perfectly matched” patients and donors.

Another important avenue for research would be to develop a theoretical basis for the methods shown here. For example, we learned that one important statistic is

the average "pseudo-reward" (i.e., the probability of a certain cycle not being used in future matchings). We believe that it may be possible to compute this quantity analytically in a simplified model with fewer types, which would then render the simulations unnecessary and possibly help decrease the gap between MYOPIC and OPT by a significant amount.

7 Conclusion

In this paper, we examined two novel algorithms for dynamic matching in a discrete-time model of kidney exchange: a *direct prediction method* that tries to predict which pair should be matched at each period; and a *multi-armed bandit method*, that uses simulations to score available exchanges in terms of their desirability. We evaluate these methods using simulations under a variety of different settings, and compared them in terms of average number of matched pairs per period to a MYOPIC algorithm that finds and immediately clears the maximal matching at each period.

We find that our *multi-armed bandit* method is able to uniformly dominate MYOPIC in all our simulation settings, including one where we draw pairs from the historical list of patients and donors that have undergone transplants in the United States.

Disclaimer

This work was supported in part by Health Resources and Services Administration contract 234-2005-37011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

8 Appendix

Environment	Statistical method	Graph statistics	Recall (TPR)	Specificity (TNR)	Precision	Accuracy
ABO	Gradient boosting	Yes	0.832	0.818	0.225	0.819
ABO	Gradient boosting	No	0.838	0.827	0.237	0.828
ABO	Logistic reg.	Yes	0.800	0.787	0.193	0.787
ABO	Logistic reg.	No	0.734	0.798	0.188	0.794
ABO	Random forests	Yes	0.360	0.983	0.577	0.946
ABO	Random forests	No	0.745	0.854	0.244	0.847
OPTN	Gradient boosting	Yes	0.597	0.828	0.435	0.786
OPTN	Gradient boosting	No	0.556	0.881	0.510	0.822
OPTN	Logistic reg.	Yes	0.730	0.553	0.266	0.585
OPTN	Logistic reg.	No	0.763	0.551	0.273	0.589
OPTN	Random forests	Yes	0.443	0.924	0.562	0.836
OPTN	Random forests	No	0.448	0.913	0.534	0.828
RSU	Gradient boosting	Yes	0.745	0.793	0.441	0.785
RSU	Gradient boosting	No	0.732	0.812	0.460	0.798
RSU	Logistic reg.	Yes	0.780	0.634	0.318	0.660
RSU	Logistic reg.	No	0.777	0.629	0.313	0.656
RSU	Random forests	Yes	0.504	0.933	0.621	0.856
RSU	Random forests	No	0.732	0.797	0.441	0.786

Table 1: Performance of statistical methods as classifiers

Three variations on *direct prediction* with different underlying statistical methods. Column GRAPH STATISTICS refers to concatenation of matrix of graph-theoretical measures for each node E^1 as well as entry and death rates E^2 . We observe that 1) inclusion of additional data E^1, E^2 does not necessarily lead to better performance; 2) No algorithm does well under all measures. Precision = $\frac{tp}{tp+fp}$, Recall = $\frac{tp}{tp+fn}$, Specificity = $\frac{tn}{tn+fp}$, Accuracy = $\frac{tp+tn}{tp+tn+fp+fn}$.

Environ.	Entry	Death	Boosting		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	1.791	0.019	1.812	0.020	-0.021	0.036	-1.173	13
		8	2.324	0.018	2.301	0.020	0.022	0.027	0.958	15
		10	2.386	0.016	2.349	0.018	0.037	0.000	1.572	16
	5	5	2.549	0.024	2.621	0.026	-0.072	0.000	-2.746	10
		8	2.870	0.018	2.880	0.020	-0.010	0.196	-0.356	16
		10	2.948	0.018	2.948	0.020	0.000	0.976	0.010	15
	7	5	3.230	0.026	3.327	0.027	-0.097	0.000	-2.903	11
		8	3.531	0.026	3.575	0.028	-0.045	0.001	-1.248	9
		10	3.512	0.026	3.555	0.028	-0.043	0.000	-1.209	9
	OPTN	3	5	2.236	0.023	2.189	0.024	0.048	0.000	2.179
8			2.314	0.016	2.259	0.017	0.055	0.000	2.431	19
10			2.348	0.017	2.322	0.018	0.026	0.007	1.112	18
5		5	2.347	0.026	2.515	0.027	-0.168	0.000	-6.679	9
		8	2.369	0.018	2.500	0.018	-0.130	0.000	-5.210	19
		10	2.393	0.017	2.519	0.018	-0.126	0.000	-5.008	21
7		5	3.043	0.028	3.373	0.030	-0.330	0.000	-9.791	10
		8	2.900	0.030	3.192	0.032	-0.292	0.000	-9.148	8
		10	2.860	0.030	3.141	0.032	-0.281	0.000	-8.957	8
RSU		3	5	2.380	0.025	2.417	0.025	-0.038	0.000	-1.553
	8		2.499	0.019	2.503	0.019	-0.004	0.552	-0.155	18
	10		2.640	0.020	2.625	0.021	0.015	0.034	0.570	16
	5	5	4.193	0.031	4.247	0.031	-0.054	0.000	-1.281	12
		8	4.134	0.027	4.140	0.027	-0.007	0.252	-0.163	15
		10	4.209	0.025	4.197	0.025	0.013	0.067	0.299	17
	7	5	5.987	0.039	6.037	0.040	-0.050	0.000	-0.836	10
		8	5.794	0.037	5.809	0.037	-0.015	0.068	-0.260	11
		10	5.924	0.043	5.918	0.043	0.006	0.334	0.097	8

Table 2: Performance of direct prediction method with gradient boosting
For selected entry and death rates.

Environ.	Entry	Death	Boosting		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	1.827	0.023	1.874	0.024	-0.047	0.004	-2.489	9
		8	2.211	0.016	2.179	0.017	0.032	0.001	1.474	19
		10	2.292	0.016	2.254	0.017	0.039	0.000	1.713	18
	5	5	2.643	0.026	2.757	0.029	-0.114	0.000	-4.131	8
		8	2.823	0.018	2.870	0.020	-0.046	0.000	-1.610	16
		10	2.915	0.020	2.914	0.021	0.000	0.937	0.011	13
	7	5	3.356	0.030	3.534	0.033	-0.178	0.000	-5.039	7
		8	3.377	0.029	3.537	0.031	-0.160	0.000	-4.537	8
		10	3.432	0.028	3.516	0.030	-0.084	0.000	-2.398	8
	OPTN	3	5	2.131	0.023	2.127	0.024	0.004	0.695	0.192
8			2.455	0.016	2.380	0.017	0.076	0.000	3.175	17
10			2.464	0.015	2.384	0.017	0.080	0.000	3.345	17
5		5	2.188	0.023	2.416	0.025	-0.228	0.000	-9.437	10
		8	2.812	0.017	2.825	0.019	-0.014	0.085	-0.485	19
		10	2.881	0.017	2.848	0.018	0.033	0.001	1.143	18
7		5	2.930	0.026	3.370	0.029	-0.440	0.000	-13.054	11
		8	3.139	0.025	3.358	0.028	-0.220	0.000	-6.543	11
		10	3.394	0.025	3.469	0.027	-0.075	0.000	-2.162	11
RSU		3	5	2.380	0.025	2.451	0.025	-0.071	0.000	-2.913
	8		2.547	0.022	2.574	0.022	-0.027	0.000	-1.044	14
	10		2.625	0.020	2.634	0.020	-0.009	0.148	-0.348	17
	5	5	4.183	0.032	4.269	0.033	-0.086	0.000	-2.021	10
		8	4.179	0.026	4.211	0.027	-0.032	0.000	-0.760	15
		10	4.164	0.025	4.183	0.025	-0.019	0.001	-0.449	16
	7	5	5.992	0.038	6.103	0.038	-0.111	0.000	-1.813	11
		8	5.832	0.036	5.870	0.036	-0.039	0.000	-0.660	12
		10	5.914	0.037	5.939	0.037	-0.024	0.005	-0.406	11

Table 3: Performance of direct prediction method with gradient boosting
Using information about graph.

Environ.	Entry	Death	Logistic		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	1.035	0.016	1.185	0.018	-0.150	0.000	-12.660	10
		8	1.479	0.015	1.530	0.015	-0.051	0.000	-3.325	19
		10	1.798	0.018	1.809	0.018	-0.011	0.158	-0.594	16
	5	5	1.774	0.024	2.029	0.026	-0.255	0.000	-12.556	8
		8	1.970	0.027	2.071	0.028	-0.101	0.000	-4.873	7
		10	2.275	0.021	2.328	0.021	-0.053	0.000	-2.284	13
	7	5	2.536	0.033	2.908	0.036	-0.372	0.000	-12.792	6
		8	2.716	0.025	2.865	0.025	-0.150	0.000	-5.219	11
		10	2.859	0.029	2.942	0.030	-0.083	0.000	-2.808	8
	OPTN	3	5	1.933	0.022	1.972	0.022	-0.039	0.004	-1.978
8			2.270	0.017	2.232	0.019	0.038	0.000	1.712	17
10			2.369	0.016	2.330	0.017	0.039	0.000	1.678	17
5		5	2.717	0.027	2.795	0.029	-0.078	0.000	-2.779	9
		8	2.893	0.019	2.881	0.019	0.013	0.027	0.442	18
		10	2.922	0.019	2.879	0.020	0.043	0.000	1.490	15
7		5	3.470	0.028	3.608	0.029	-0.138	0.000	-3.832	11
		8	3.602	0.025	3.640	0.026	-0.038	0.000	-1.049	12
		10	3.611	0.026	3.606	0.027	0.005	0.470	0.148	11
RSU	3	5	2.037	0.025	2.434	0.027	-0.397	0.000	-16.302	9
		8	2.221	0.034	2.331	0.035	-0.109	0.000	-4.688	5
		10	2.355	0.019	2.399	0.019	-0.044	0.000	-1.825	17
	5	5	3.636	0.030	4.249	0.033	-0.613	0.000	-14.429	10
		8	3.886	0.028	4.080	0.028	-0.194	0.000	-4.752	14
		10	3.966	0.026	4.031	0.025	-0.066	0.000	-1.634	16
	7	5	5.164	0.039	6.040	0.042	-0.877	0.000	-14.512	9
		8	5.635	0.038	5.875	0.039	-0.240	0.000	-4.080	10
		10	5.606	0.038	5.706	0.038	-0.100	0.000	-1.751	10

Table 4: Performance of direct prediction method with logistic regression
Not using information about graph.

Environ.	Entry	Death	Logistic		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	1.063	0.018	1.196	0.019	-0.133	0.000	-11.114	9
		8	1.350	0.017	1.405	0.017	-0.055	0.000	-3.891	14
		10	1.638	0.019	1.658	0.019	-0.020	0.001	-1.186	14
	5	5	1.894	0.023	2.047	0.024	-0.153	0.000	-7.452	9
		8	2.170	0.019	2.224	0.019	-0.054	0.000	-2.445	15
		10	2.430	0.019	2.439	0.019	-0.010	0.109	-0.404	17
	7	5	2.745	0.034	2.963	0.035	-0.218	0.000	-7.370	6
		8	2.991	0.032	3.033	0.032	-0.043	0.001	-1.403	7
		10	3.257	0.028	3.265	0.029	-0.008	0.290	-0.235	9
	OPTN	3	5	1.803	0.021	1.846	0.022	-0.043	0.025	-2.313
8			2.302	0.018	2.263	0.019	0.039	0.000	1.733	16
10			2.362	0.016	2.304	0.018	0.058	0.000	2.519	16
5		5	2.519	0.027	2.642	0.028	-0.123	0.000	-4.664	9
		8	2.925	0.019	2.896	0.020	0.028	0.003	0.984	16
		10	2.956	0.018	2.908	0.019	0.048	0.000	1.643	18
7		5	3.192	0.027	3.459	0.029	-0.267	0.000	-7.716	11
		8	3.604	0.028	3.604	0.029	-0.000	0.940	-0.009	10
		10	3.654	0.024	3.617	0.025	0.037	0.000	1.014	13
RSU	3	5	2.062	0.023	2.424	0.024	-0.362	0.000	-14.916	11
		8	2.164	0.021	2.296	0.021	-0.132	0.000	-5.760	14
		10	2.284	0.021	2.348	0.021	-0.065	0.000	-2.762	15
	5	5	3.660	0.035	4.215	0.037	-0.555	0.000	-13.169	8
		8	3.876	0.028	4.061	0.027	-0.186	0.000	-4.572	15
		10	3.883	0.026	3.979	0.025	-0.096	0.000	-2.424	17
	7	5	5.177	0.054	5.992	0.062	-0.815	0.000	-13.596	4
		8	5.638	0.046	5.835	0.044	-0.197	0.000	-3.376	8
		10	5.688	0.043	5.787	0.041	-0.100	0.000	-1.727	9

Table 5: Performance of direct prediction method with logistic regression
Using information about graph.

Environ.	Entry	Death	Forest		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	1.173	0.019	1.240	0.020	-0.067	0.000	-5.417	8
		8	1.331	0.016	1.373	0.017	-0.042	0.000	-3.082	14
		10	1.513	0.017	1.539	0.017	-0.026	0.002	-1.672	15
	5	5	1.952	0.025	2.049	0.026	-0.097	0.000	-4.733	8
		8	2.223	0.019	2.281	0.020	-0.059	0.000	-2.573	15
		10	2.366	0.018	2.396	0.019	-0.030	0.000	-1.240	17
	7	5	2.805	0.030	2.913	0.031	-0.108	0.000	-3.715	8
		8	2.888	0.029	2.978	0.030	-0.089	0.000	-3.002	8
		10	2.990	0.025	3.049	0.026	-0.059	0.000	-1.936	11
	OPTN	3	5	0.962	0.018	1.256	0.020	-0.293	0.000	-23.349
8			0.815	0.011	1.169	0.014	-0.354	0.000	-30.298	19
10			0.775	0.012	1.168	0.014	-0.394	0.000	-33.693	17
5		5	1.566	0.025	2.185	0.030	-0.619	0.000	-28.324	7
		8	1.269	0.014	1.969	0.018	-0.701	0.000	-35.585	17
		10	1.160	0.014	1.895	0.018	-0.735	0.000	-38.766	16
7		5	2.322	0.025	3.275	0.029	-0.953	0.000	-29.110	10
		8	1.924	0.024	2.993	0.029	-1.069	0.000	-35.731	9
		10	1.736	0.024	2.807	0.030	-1.071	0.000	-38.162	8
RSU	3	5	2.423	0.026	2.465	0.027	-0.043	0.000	-1.728	9
		8	2.670	0.022	2.664	0.022	0.006	0.368	0.213	14
		10	2.752	0.020	2.733	0.020	0.019	0.002	0.685	17
	5	5	4.175	0.037	4.214	0.037	-0.039	0.008	-0.927	8
		8	4.163	0.025	4.172	0.026	-0.009	0.145	-0.213	16
		10	4.288	0.027	4.283	0.027	0.005	0.499	0.125	15
	7	5	5.996	0.052	6.042	0.052	-0.047	0.013	-0.772	6
		8	5.915	0.041	5.909	0.041	0.006	0.455	0.102	9
		10	5.827	0.036	5.820	0.036	0.007	0.481	0.121	11

Table 6: Performance of direct prediction method with random forests
Not using information about graph.

Environ.	Entry	Death	Forest		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3	5	0.953	0.017	1.181	0.018	-0.228	0.000	-19.284	9
		8	0.934	0.013	1.152	0.014	-0.217	0.000	-18.874	15
		10	0.916	0.013	1.118	0.014	-0.202	0.000	-18.084	15
	5	5	1.719	0.025	2.066	0.027	-0.347	0.000	-16.805	7
		8	1.645	0.017	1.938	0.018	-0.293	0.000	-15.105	16
		10	1.673	0.018	1.940	0.019	-0.268	0.000	-13.796	14
	7	5	2.385	0.059	2.883	0.061	-0.497	0.049	-17.253	2
		8	2.405	0.031	2.811	0.032	-0.406	0.000	-14.460	7
		10	2.424	0.027	2.819	0.028	-0.394	0.000	-13.983	9
	OPTN	3	5	0.517	0.012	1.180	0.018	-0.663	0.000	-56.214
8			0.552	0.010	1.087	0.014	-0.535	0.000	-49.181	17
10			0.619	0.010	1.111	0.013	-0.491	0.000	-44.245	18
5		5	0.946	0.018	2.161	0.027	-1.215	0.000	-56.221	8
		8	0.909	0.012	1.962	0.018	-1.053	0.000	-53.661	17
		10	0.963	0.012	1.873	0.017	-0.910	0.000	-48.590	17
7		5	2.173	0.028	3.313	0.032	-1.141	0.000	-34.424	9
		8	1.314	0.017	2.993	0.026	-1.679	0.000	-56.088	12
		10	1.344	0.017	2.822	0.025	-1.479	0.000	-52.392	12
RSU	3	5	2.102	0.025	2.372	0.027	-0.270	0.000	-11.369	9
		8	2.074	0.019	2.304	0.020	-0.229	0.000	-9.957	15
		10	2.087	0.018	2.301	0.019	-0.215	0.000	-9.322	16
	5	5	3.859	0.032	4.243	0.033	-0.385	0.000	-9.064	10
		8	3.744	0.025	4.070	0.026	-0.325	0.000	-7.998	15
		10	3.666	0.020	3.976	0.021	-0.311	0.000	-7.815	23
	7	5	5.564	0.052	6.027	0.051	-0.464	0.000	-7.693	6
		8	5.455	0.041	5.821	0.043	-0.365	0.000	-6.277	8
		10	5.427	0.042	5.762	0.043	-0.335	0.000	-5.816	8

Table 7: Performance of direct prediction method with random forests
Using information about graph.

Environ.	Entry	Death	UCB1		Myopic		Difference	p-value	Ratio (%)	N
			Mean	Std Error	Mean	Std Error				
ABO	3.0	5.0	1.156	0.021	1.142	0.021	0.014	0.013	1.263	5
		8.0	1.121	0.026	1.116	0.026	0.005	0.369	0.418	3
		10.0	1.095	0.020	1.079	0.020	0.016	0.003	1.484	5
	5.0	5.0	2.035	0.009	2.017	0.009	0.018	0.000	0.875	63
		8.0	2.021	0.031	1.996	0.031	0.025	0.004	1.237	4
		10.0	1.967	0.025	1.945	0.025	0.022	0.001	1.133	6
	7.0	5.0	2.908	0.050	2.894	0.050	0.014	0.048	0.497	8
		8.0	2.831	0.027	2.805	0.026	0.026	0.000	0.911	12
		10.0	2.801	0.033	2.771	0.033	0.030	0.005	1.088	5
OPTN	3.0	5.0	1.196	0.020	1.160	0.020	0.036	0.000	3.107	6
		8.0	1.063	0.027	1.043	0.027	0.020	0.023	1.919	3
		10.0	0.998	0.020	0.971	0.019	0.027	0.002	2.763	5
	5.0	5.0	2.260	0.039	2.175	0.038	0.085	0.001	3.917	4
		8.0	1.981	0.025	1.914	0.025	0.066	0.000	3.469	6
		10.0	1.868	0.030	1.825	0.030	0.043	0.012	2.361	4
	7.0	5.0	3.398	0.020	3.264	0.020	0.134	0.000	4.106	34
		8.0	3.071	0.026	2.969	0.026	0.102	0.000	3.447	9
		10.0	2.880	0.037	2.790	0.037	0.090	0.000	3.225	5
RSU	3.0	5.0	2.477	0.026	2.398	0.026	0.078	0.000	3.268	8
		8.0	2.380	0.039	2.310	0.038	0.069	0.005	3.004	3
		10.0	2.268	0.029	2.201	0.029	0.067	0.000	3.057	5
	5.0	5.0	4.330	0.025	4.201	0.025	0.128	0.000	3.056	40
		8.0	4.190	0.040	4.072	0.040	0.118	0.000	2.900	8
		10.0	4.084	0.011	3.973	0.011	0.112	0.000	2.815	64
	7.0	5.0	5.519	0.370	5.370	0.399	0.148	nan	2.759	1
		8.0	6.041	0.044	5.870	0.044	0.171	0.000	2.906	19
		10.0	5.904	0.023	5.742	0.023	0.162	0.000	2.824	53

Table 8: Multi-armed bandit algorithm UCB1

ENVIRON is the environment used for simulation. ENTRY and DEATH are the Poisson entry rate of entry and the the Geometric rate of departure (times 100), respectively. MEAN and STD refers to the average number of matched patients over 1000 periods. DIFFERENCE and RATIO compare the average improvement between the algorithm and MYOPIC. N is the number of 1000-period simulations that used that particular configuration. Tables for other bandits are similar, and can be found in the online supplement that will be available online.

Env.	Algo.	Entry	Death	MAB Mean	SEM	Myopic Mean	SEM	Difference	p-value	Improvement (%)	N
ABO	Thompson	3	0.03	1.220	0.018	1.209	0.018	0.010	0.002	0.864	9
ABO	Thompson	3	0.05	1.184	0.011	1.170	0.011	0.014	$< 10^{-3}$	1.204	10
ABO	Thompson	3	0.07	1.144	0.003	1.130	0.003	0.014	$< 10^{-3}$	1.258	220
ABO	Thompson	5	0.03	2.081	0.05	2.059	0.05	0.022	0.025	1.088	6
ABO	Thompson	5	0.05	2.05	0.020	2.032	0.020	0.019	$< 10^{-3}$	0.914	14
ABO	Thompson	5	0.07	2.015	0.011	1.999	0.011	0.016	$< 10^{-3}$	0.820	33
ABO	Thompson	7	0.03	2.894	0.062	2.870	0.062	0.023	0.025	0.812	12
ABO	Thompson	7	0.05	2.893	0.015	2.875	0.015	0.018	$< 10^{-3}$	0.640	63
ABO	Thompson	7	0.07	2.896	0.027	2.877	0.027	0.019	$< 10^{-3}$	0.652	9
ABO	UCB1	3	0.03	1.244	0.016	1.231	0.016	0.012	$< 10^{-3}$	0.990	10
ABO	UCB1	3	0.05	1.171	0.011	1.156	0.011	0.015	$< 10^{-3}$	1.295	13
ABO	UCB1	3	0.07	1.150	0.003	1.136	0.003	0.014	$< 10^{-3}$	1.218	211
ABO	UCB1	5	0.03	2.109	0.046	2.087	0.046	0.022	0.049	1.054	6
ABO	UCB1	5	0.05	2.042	0.006	2.025	0.006	0.017	$< 10^{-3}$	0.841	108
ABO	UCB1	5	0.07	2.025	0.011	2.003	0.011	0.022	$< 10^{-3}$	1.099	33
ABO	UCB1	7	0.03	2.856	0.062	2.837	0.062	0.019	0.076	0.655	11
ABO	UCB1	7	0.05	2.921	0.016	2.898	0.016	0.024	$< 10^{-3}$	0.815	48
ABO	UCB1	7	0.07	2.872	0.026	2.842	0.027	0.03	$< 10^{-3}$	1.054	9

Table 9: Multi-armed bandit algorithm Thompson Sampling

ENTRY and DEATH are the Poisson entry rate of entry and the Geometric rate of departure. MEAN and STD refers to the average number of matched patients over 1000 periods. DIFFERENCE and RATIO compare the average improvement between the algorithm and MYOPIC. N is the number of 1000-period simulations that used that particular configuration.

Env.	Algo.	Entry	Death	MAB Mean	SEM	Myopic Mean	SEM	Difference	p-value	Improvement (%)	N
RSU	Thompson	3	0.03	2.610	0.038	2.529	0.037	0.081	$< 10^{-3}$	3.210	5
RSU	Thompson	3	0.05	2.448	0.018	2.364	0.017	0.084	$< 10^{-3}$	3.573	8
RSU	Thompson	3	0.07	2.408	0.007	2.345	0.007	0.064	$< 10^{-3}$	2.711	85
RSU	Thompson	5	0.03	4.391	0.079	4.263	0.079	0.128	$< 10^{-3}$	2.998	5
RSU	Thompson	5	0.05	4.325	0.014	4.194	0.014	0.131	$< 10^{-3}$	3.118	62
RSU	Thompson	5	0.07	4.240	0.027	4.123	0.027	0.117	$< 10^{-3}$	2.846	12
RSU	Thompson	7	0.03	6.542	0.251	6.262	0.258	0.280	0.042	4.478	4
RSU	Thompson	7	0.05	6.118	0.037	5.950	0.037	0.168	$< 10^{-3}$	2.823	55
RSU	Thompson	7	0.07	6.060	0.057	5.886	0.057	0.174	$< 10^{-3}$	2.951	7
RSU	UCB1	3	0.03	2.626	0.038	2.562	0.038	0.064	$< 10^{-3}$	2.495	5
RSU	UCB1	3	0.05	2.475	0.019	2.397	0.018	0.079	$< 10^{-3}$	3.279	8
RSU	UCB1	3	0.07	2.398	0.007	2.335	0.007	0.063	$< 10^{-3}$	2.692	85
RSU	UCB1	5	0.03	4.523	0.063	4.401	0.063	0.122	$< 10^{-3}$	2.775	12
RSU	UCB1	5	0.05	4.333	0.014	4.209	0.013	0.124	$< 10^{-3}$	2.952	65
RSU	UCB1	5	0.07	4.244	0.027	4.134	0.027	0.111	$< 10^{-3}$	2.673	12
RSU	UCB1	7	0.03	6.023	0.157	5.864	0.161	0.159	0.010	2.716	7
RSU	UCB1	7	0.05	6.141	0.037	5.962	0.037	0.179	$< 10^{-3}$	3.004	50
RSU	UCB1	7	0.07	5.995	0.056	5.856	0.057	0.139	$< 10^{-3}$	2.382	7

Table 10: Multi-armed bandit algorithm in RSU environment

ENTRY and DEATH are the Poisson entry rate of entry and the the Geometric rate of departure. MEAN and STD refers to the average number of matched patients over 1000 periods. DIFFERENCE and RATIO compare the average improvement between the algorithm and MYOPIC. N is the number of 1000-period simulations that used that particular configuration.

Algo.	Entry Mean	Death Std Error	MAB Mean	Myopic Std Error	Difference	p-value	Improvement (%)	N		
Thompson	3	0.010	1.664	0.020	1.593	0.019	0.072	$< 10^{-3}$	4.517	10
Thompson	3	0.030	1.347	0.010	1.297	0.010	0.050	$< 10^{-3}$	3.816	24
Thompson	3	0.050	1.198	0.014	1.162	0.014	0.036	$< 10^{-3}$	3.098	6
Thompson	3	0.070	1.078	0.003	1.052	0.003	0.026	$< 10^{-3}$	2.434	283
Thompson	3	0.090	1.002	0.002	0.981	0.002	0.021	$< 10^{-3}$	2.124	361
Thompson	3	0.100	0.999	0.018	0.979	0.018	0.021	0.031	2.112	3
Thompson	4	0.010	2.280	0.032	2.174	0.031	0.107	$< 10^{-3}$	4.900	6
Thompson	4	0.030	1.894	0.014	1.822	0.014	0.072	$< 10^{-3}$	3.928	18
Thompson	4	0.050	1.721	0.015	1.668	0.015	0.053	$< 10^{-3}$	3.165	15
Thompson	4	0.070	1.570	0.003	1.527	0.003	0.043	$< 10^{-3}$	2.812	260
Thompson	4	0.090	1.466	0.003	1.429	0.002	0.036	$< 10^{-3}$	2.552	465
Thompson	4	0.100	1.416	0.004	1.382	0.004	0.034	$< 10^{-3}$	2.437	211
Thompson	5	0.010	2.879	0.051	2.756	0.050	0.123	0.002	4.475	4
Thompson	5	0.030	2.495	0.021	2.402	0.020	0.092	$< 10^{-3}$	3.850	13
Thompson	5	0.050	2.237	0.034	2.163	0.034	0.074	0.066	3.406	4
Thompson	5	0.070	2.075	0.010	2.009	0.010	0.066	$< 10^{-3}$	3.275	38
Thompson	5	0.090	1.930	0.014	1.875	0.014	0.055	$< 10^{-3}$	2.928	20
Thompson	5	0.100	1.920	0.018	1.867	0.017	0.053	$< 10^{-3}$	2.838	6
Thompson	6	0.010	3.677	0.091	3.472	0.085	0.206	0.029	5.920	3
Thompson	6	0.030	3.079	0.028	2.962	0.027	0.117	$< 10^{-3}$	3.955	9
Thompson	6	0.050	2.769	0.027	2.675	0.026	0.094	$< 10^{-3}$	3.514	8
Thompson	6	0.070	2.623	0.011	2.537	0.011	0.086	$< 10^{-3}$	3.387	40
Thompson	6	0.090	2.450	0.010	2.379	0.010	0.071	$< 10^{-3}$	2.976	48
Thompson	6	0.100	2.357	0.018	2.295	0.018	0.062	$< 10^{-3}$	2.721	14
Thompson	7	0.010	4.108	0.137	3.900	0.135	0.208	0.005	5.339	3
Thompson	7	0.030	3.673	0.038	3.515	0.037	0.158	$< 10^{-3}$	4.493	6
Thompson	7	0.050	3.396	0.009	3.265	0.009	0.132	$< 10^{-3}$	4.030	92
Thompson	7	0.070	3.124	0.028	3.026	0.027	0.097	$< 10^{-3}$	3.213	8
Thompson	7	0.090	2.921	0.024	2.836	0.024	0.084	$< 10^{-3}$	2.976	10
Thompson	7	0.100	2.910	0.022	2.807	0.022	0.103	$< 10^{-3}$	3.680	6
Thompson	8	0.030	4.398	0.136	4.181	0.128	0.217	$< 10^{-3}$	5.195	1
Thompson	8	0.070	3.732	0.015	3.600	0.015	0.132	$< 10^{-3}$	3.670	35
Thompson	8	0.090	3.506	0.013	3.391	0.013	0.115	$< 10^{-3}$	3.391	41

Table 11: Multi-armed bandit algorithm performance in OPTN environment
ENTRY and DEATH are the Poisson entry rate of entry and the the Geometric rate of departure. MEAN and STD refers to the average number of matched patients over 1000 periods. DIFFERENCE and RATIO compare the average improvement between the algorithm and MYOPIC. N is the number of 1000-period simulations that used that particular configuration.

Algo.	Entry Mean	Death Std Error	MAB Mean	Myopic Std Error	Difference	p-value	Improvement (%)	N		
UCB1	3	0.010	1.640	0.018	1.579	0.018	0.061	$< 10^{-3}$	3.866	11
UCB1	3	0.030	1.343	0.010	1.302	0.010	0.041	$< 10^{-3}$	3.168	26
UCB1	3	0.050	1.195	0.014	1.159	0.014	0.036	$< 10^{-3}$	3.106	6
UCB1	3	0.070	1.087	0.003	1.061	0.003	0.027	$< 10^{-3}$	2.504	306
UCB1	3	0.090	1.004	0.002	0.983	0.002	0.021	$< 10^{-3}$	2.178	396
UCB1	3	0.100	0.997	0.014	0.970	0.014	0.027	0.002	2.763	5
UCB1	4	0.010	2.275	0.030	2.190	0.029	0.085	$< 10^{-3}$	3.881	7
UCB1	4	0.030	1.902	0.014	1.839	0.014	0.064	$< 10^{-3}$	3.458	19
UCB1	4	0.050	1.702	0.014	1.647	0.014	0.055	$< 10^{-3}$	3.328	17
UCB1	4	0.070	1.570	0.003	1.527	0.003	0.043	$< 10^{-3}$	2.794	436
UCB1	4	0.090	1.457	0.003	1.422	0.003	0.035	$< 10^{-3}$	2.470	336
UCB1	4	0.100	1.419	0.004	1.387	0.004	0.032	$< 10^{-3}$	2.333	168
UCB1	5	0.010	2.907	0.051	2.776	0.050	0.131	$< 10^{-3}$	4.726	5
UCB1	5	0.030	2.489	0.019	2.395	0.019	0.094	$< 10^{-3}$	3.945	14
UCB1	5	0.050	2.260	0.027	2.174	0.027	0.085	0.001	3.917	4
UCB1	5	0.070	2.080	0.010	2.022	0.010	0.058	$< 10^{-3}$	2.876	38
UCB1	5	0.090	1.942	0.008	1.893	0.008	0.050	$< 10^{-3}$	2.627	63
UCB1	5	0.100	1.866	0.021	1.823	0.021	0.043	0.012	2.360	4
UCB1	6	0.010	3.521	0.085	3.354	0.086	0.167	0.010	4.984	3
UCB1	6	0.030	3.077	0.027	2.956	0.026	0.122	$< 10^{-3}$	4.112	11
UCB1	6	0.050	2.772	0.024	2.683	0.024	0.089	$< 10^{-3}$	3.322	10
UCB1	6	0.070	2.632	0.011	2.544	0.011	0.088	$< 10^{-3}$	3.452	41
UCB1	6	0.090	2.451	0.010	2.376	0.009	0.075	$< 10^{-3}$	3.158	54
UCB1	6	0.100	2.368	0.018	2.302	0.018	0.065	$< 10^{-3}$	2.826	15
UCB1	7	0.010	4.047	0.151	3.875	0.144	0.172	0.017	4.447	3
UCB1	7	0.030	3.653	0.033	3.525	0.033	0.129	$< 10^{-3}$	3.648	9
UCB1	7	0.050	3.400	0.009	3.267	0.009	0.133	$< 10^{-3}$	4.064	85
UCB1	7	0.070	3.183	0.026	3.079	0.026	0.104	$< 10^{-3}$	3.378	9
UCB1	7	0.090	2.959	0.023	2.867	0.023	0.093	$< 10^{-3}$	3.241	11
UCB1	7	0.100	2.890	0.026	2.797	0.025	0.093	$< 10^{-3}$	3.321	6
UCB1	8	0.030	4.315	0.116	4.128	0.110	0.186	$< 10^{-3}$	4.510	1
UCB1	8	0.070	3.720	0.015	3.591	0.015	0.129	$< 10^{-3}$	3.591	37
UCB1	8	0.090	3.489	0.014	3.379	0.014	0.110	$< 10^{-3}$	3.263	35
UCB1	10	0.050	5.174	0.083	4.946	0.081	0.228	$< 10^{-3}$	4.610	7

Table 12: Multi-armed bandit algorithm performance in OPTN environment (contd)

ENTRY and DEATH are the Poisson entry rate of entry and the the Geometric rate of departure. MEAN and STD refers to the average number of matched patients over 1000 periods. DIFFERENCE and RATIO compare the average improvement between the algorithm and MYOPIC. N is the number of 1000-period simulations that used that particular configuration.

	Entry	Death	Adj. Death	Closest	ImprovementAtClosest
2	3.000	0.030	0.010	0.010	3.866
4	3.000	0.050	0.017	0.020	3.602
6	3.000	0.070	0.024	0.020	3.602
12	4.000	0.030	0.008	0.010	3.881
14	4.000	0.050	0.013	0.010	3.881
16	4.000	0.070	0.018	0.020	4.297
22	5.000	0.030	0.006	0.010	4.726
24	5.000	0.050	0.010	0.010	4.726
26	5.000	0.070	0.014	0.010	4.726
32	6.000	0.030	0.005	0.010	4.984
34	6.000	0.050	0.009	0.010	4.984
36	6.000	0.070	0.012	0.010	4.984
44	7.000	0.050	0.007	0.010	4.447
46	7.000	0.070	0.010	0.010	4.447
61	3.000	0.030	0.010	0.010	3.866
63	3.000	0.050	0.017	0.020	3.602
65	3.000	0.070	0.024	0.020	3.602
71	4.000	0.030	0.008	0.010	3.881
73	4.000	0.050	0.013	0.010	3.881
75	4.000	0.070	0.018	0.020	4.297
81	5.000	0.030	0.006	0.010	4.726
83	5.000	0.050	0.010	0.010	4.726
85	5.000	0.070	0.014	0.010	4.726
91	6.000	0.030	0.005	0.010	4.984
93	6.000	0.050	0.009	0.010	4.984
95	6.000	0.070	0.012	0.010	4.984
103	7.000	0.050	0.007	0.010	4.447
105	7.000	0.070	0.010	0.010	4.447

Table 13: Approximate comparison to a *chunk-matching* algorithm

Rough approximation of our MAB method performance to a *chunk-matching* algorithm (see text for details). ENTRY and DEATH are entry and death rates in the original environment; ADJ. DEATH is an adjusted death rate to take into account that only one pair enters per period; CLOSEST is the closest death rate we have in the original environment; IMPROVEMENTATCLOSEST refers to the improvement at the closest death rate in the original environment.

Part II

Econometrics

Heterogenous Production Functions, Panel Data, and Productivity Dispersion

JEREMY FOX, VITOR HADAD, STEFAN HODERLEIN

AMIL PETRIN, ROBERT SHERMAN

1 Introduction

Appropriately modeling unobserved heterogeneity is one of the foremost challenges faced by econometricians. In the context of panel data, it has become standard to controlling for unobserved heterogeneity via a *fixed effects* approach, where we do not impose restrictions the distribution of latent heterogeneity parameters conditional on observable characteristics Wooldridge (2010). However, the most ubiquitous fixed effects strategy in panel data is via the introduction of a time-invariant random intercept (Arellano, 2003), and there are many instances of economic problems where this is might not be appropriate (Browning et al., 2007).

Here, we present constructive identification and estimation results for the moments

and marginal distribution of random coefficients in a linear panel data model with *two* random coefficients

$$Y_{it} = A_{it} + X_{it}B_{it} + Z_{it}^T\beta_t \quad (1)$$

where the intercept A_{it} and slope B_{it} follow a specific autoregressive process but are otherwise allowed to be arbitrarily correlated with all regressors, and Z_{it} is a (potentially large) vector of covariates associated with individual-invariant coefficients β .

Our results develop upon and extend the results of earlier work by Graham and Powell (2012), who only focus on the first moments of the random coefficients and have stricter timing assumptions than ours. Another paper is Arellano and Bonhomme (2011), who, like us, provide a method for identification and estimation of higher moments and distributional characteristics of random coefficients, but they maintain certain regularity assumptions that we do not impose. Our model also permits that lagged dependent variables to be regressors. More broadly, this paper contributes to the larger literature of panel data models with unobserved heterogeneity, as we discuss below.

In our particular application, we take Y_{it} to be value-added log-production, while the two random coefficients represent total factor productivity (TFP) and the elasticity of one of the inputs in a Cobb-Douglas model of production. There is strong evidence for heterogeneity in this setting, where it is known as *productivity dispersion*. In the United States, Syverson (2004) finds large plant-level discrepancies in TFP, noting that within four-digit SIC industries “the plant at the 90th percentile of the productivity distribution makes almost twice as much output with the same measured inputs as the 10th percentile plant”, and a later survey by Syverson (2011) confirms this to be a typical finding. In fact, other work by (Hsieh and Klenow,

2009) indicates that this gap may be small compared to other countries. In our work we note, however, that the large observed variation in TFP might be a product of restrictive statistical models: when only the intercept is able to vary, it may end up absorbing most the heterogeneity in the data, leading to inflated estimates of the standard deviation. On the other hand, we remark that our specific model and data differ from the ones above so we cannot directly compare our results, but leave it as a suggestion for future research.

2 Background and literature review

Panel data and heterogeneity In full generality, panel data models can be represented as

$$Y_{it} = m(H_{it}, X_{it}, U_{it}) \quad (2)$$

where Y_{it} is the dependent variable of interest for unit i at period t , H_{it} is an individual-specific, possibly infinite-dimensional object often called the *unobserved heterogeneity* component, X_{it} are observed characteristics, and U_{it} is an idiosyncratic error term. For example, Y_{it} might represent an individual's wages over time, X_{it} a vector of characteristics that includes educational attainment, H_{it} a measure of their cognitive or manual ability, and U_{it} measurement error. The econometrician is usually interested in some functional of this object, such as the average effect of a change in X_{it} on Y_t while keeping everything else fixed – known as the *average partial effect* (APE) (Blundell and Powell, 2003).

Particular care must be taken with the *unobserved heterogeneity* parameter H_{it} . Browning et al. (2007, p.2) define *heterogeneity* as “the dispersion in factors that are relevant and known to individual agents when making a particular decision”¹. The

¹Browning *et al* were, in turn, paraphrasing Cunha et al. (2005)

qualifier *unobserved* serves to emphasize that the econometrician does not have access to information about such relevant factors. This is problematic because, even when H_{it} itself is not of interest, it may preclude the identification of causal parameters. To continue our running example: even if we observe that workers with higher educational attainment receive higher wages, we cannot a priori claim that we have found a causal effect of X_{it} , as it may be that high-ability (H_{it}) workers are able to self-select into higher-paying jobs.

One of the foremost advantages of dealing with panel data is that, under certain conditions, the econometrician may be able to leverage the existence of multiple observations per unit of analysis to control for this unobserved heterogeneity. This is possible, for example, if the researcher additionally imposes that m is linear, and that all the unobserved heterogeneity can be explained by an individual but time-invariant shift, as in:

$$Y_{it} = A_i + X_{it}^T \beta + U_{it} \quad (3)$$

where all the heterogeneity H_{it} is entirely subsumed in the random scalar intercept coefficient A_i . Such linear panel data models are the econometric “workhorse in empirical studies” Wooldridge (2005).

The average partial effect in this case is represented by components of the shared coefficient β , which can be identified under additional restrictions on the stochastic process governing U_{it} .

In common econometric parlance, identification strategies that allow for $Cov(X_{it}, A_{it}) \neq 0$ are named *fixed effects* models, while those that additionally impose that this covariance must be zero are called *random effect* models. Arellano (2003); Wooldridge (2010). More generally, Graham and Powell (2012) define the class of *fixed effects* methods as one that implies restrictions on the functional form of m and on the

conditional distribution of the idiosyncratic shocks $F(U_{it}|X., A.)$, but leaves the conditional distribution of unobserved heterogeneity $F(A_{it}|X_{it})$ unrestricted. A stricter class of *random effects* methods additionally imposes an independence assumption between $A.$ and $X.$. While facilitating identification and estimation, the latter is harder to motivate economically – it would mean, in our example, that workers’ wages were unrelated to their unobserved ability (see Griliches (1977) for a discussion).

Random coefficients In Equation 3, all the heterogeneity is subsumed into one random scalar that enters the model additively. While such models are extremely common in applied work, their ubiquitousness may be explained more for their convenience than for their economic realism. As put by Browning et al. (2007, p.11),

Almost always decisions on how to include allowance for heterogeneity are made using conventional schemes that have been designed by statisticians to put in the heterogeneity in such a way that we can immediately take it out again.

Therefore, in this paper we focus on partially rolling back some of the assumptions above and allow for a more general case with more than one correlated random coefficient.

We will be working with a specific version of this general *random coefficients* model

$$Y_{it} = A_{it} + X_{it}^T B_{it} + Z_{it}^T \beta \quad (4)$$

where now $H_{it} = [A_{it}, B_{it}^T]^T$, so that now we have *vector-valued heterogeneity*. Moreover, we will be principally interest with the case when A_{it}, B_{it} are *endogenous*, that is, not independent of X_{it} (in the opposite case we say that the random coefficients are *exogenous*).

The econometric literature dealing with this more general case is substantially

smaller. Early work by Swamy (1970) studied a panel data model with exogenous, stationary random coefficients and produced a feasible GLS estimator for random coefficient averages. Mundlak (1978) and Chamberlain (1984) were one of the first to study panel data models with correlated random coefficients and derived efficient estimators under parametric assumptions about the distribution of random coefficients and errors. Chamberlain (1992) derived an efficient semiparametric estimator of the first moment of random coefficient under regularity assumptions. Later, Arellano and Bonhomme (2011) produced identification results for higher moments of the distribution of random coefficients under similar conditions, while Graham and Powell (2012) extended his approach to obtain the first moment in the *irregular* case when such assumptions are not satisfied. As we will see, the main feature of irregular identification is convergence rates smaller than root-n (Lewbel, 2016), which in turn happens due to the dependence on aspects of “small” subpopulations defined on measure-zero sets.

It should also be noted that even without panel data, there may be instances where aspects of random coefficients may be identified by if the variation in X_{it} can be explained by variation in a set exogenous excluded variables. Examples of such *triangular* models include Heckman and Vytlacil (1998), Hoderlein et al. (2010), and Hoderlein et al. (2017).

Production function estimation Our illustrative empirical application will involve the identification and estimation of a production function, so that Y_{it} will represent output and X_{it} inputs to production. This is a vast and comprehensive field on its own, so here we will simply note that two issues are pervasive throughout the literature and relevant to our paper.

First, there is a need to take heterogeneity into account when estimating production functions. The recognition of this fact goes at least as far back as Marschak and

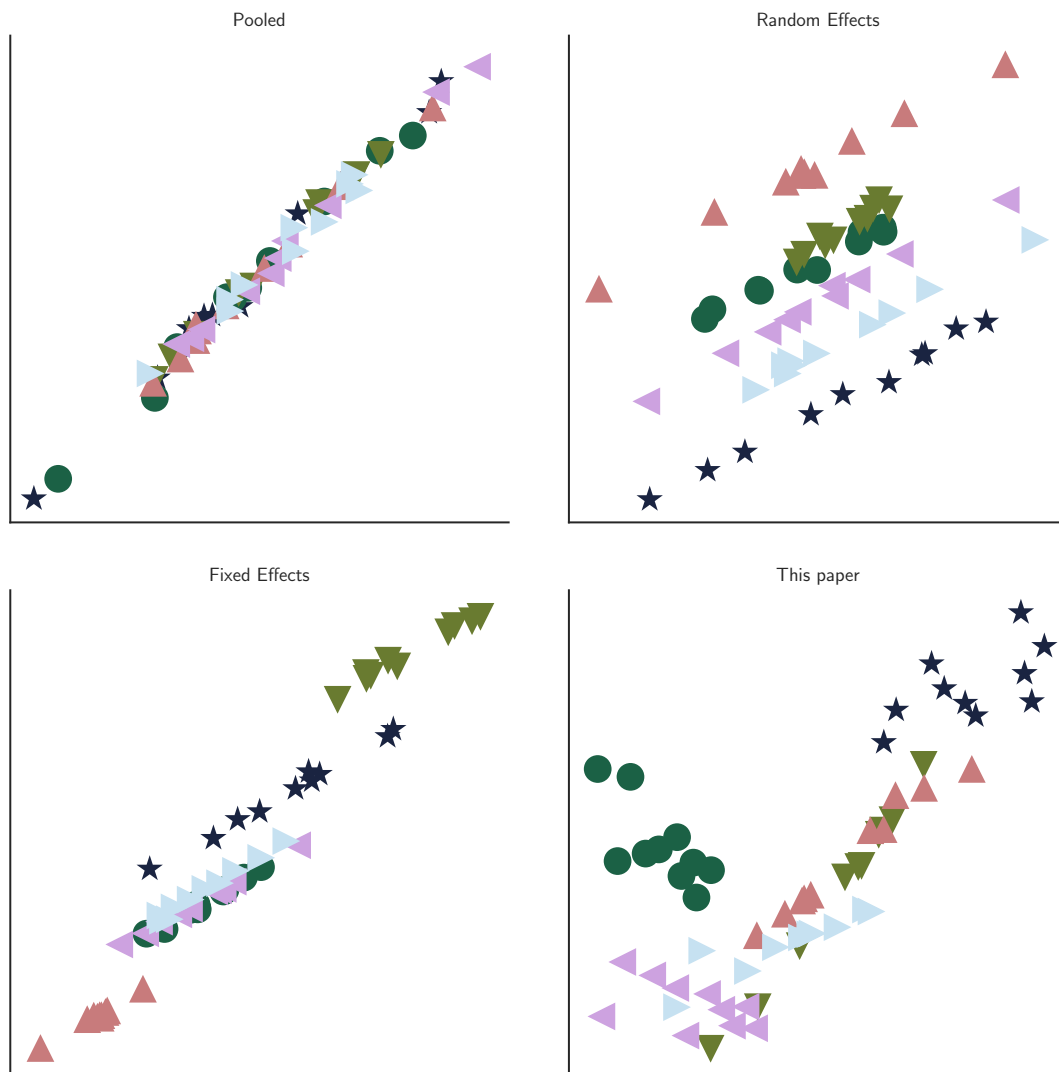


Figure 12: Linear panel data models

Different assumptions leading to different panel data models. Each different marker type and color indicates a different individual. *Pooled*: all observations share same intercept and slope; *“Random effects” random intercept*: each observation possesses their own time-invariant slope, but all share the same slope; *“Fixed effects” random intercept*: same as random effects, but intercept and regressors can be correlated (figure shows positive correlation); *This paper*: we generalize the model by 1) allowing for intercept and slope to be correlated with regressors, and 2) allowing both intercept and slope to drift according to a specific Markovian process.

Andrews (1944) who note that [p. 145]

[T]he production function will change, even within the same industry, from firm to firm and from year to year, depending on the technical knowledge, the will, effort, and luck of a given entrepreneur: these factors (...) may be represented by one or more random parameters.

Second, that this heterogeneity is plausibly endogenous, since a firm's choice of inputs is likely correlated with their own (unobserved) productivity (Gandhi et al., 2011). Empirical applications usually address this issue via a dynamic panel approach (as in Arellano and Bond (1991)) or via proxy variable approaches (as in Levinsohn and Petrin (2003)).

In allowing for correlated random coefficients, our paper incorporates a flexible form of vector-valued heterogeneity that also takes the endogeneity problem into account. The next sections will sketch our general argument.

3 Model and assumptions

From now on, in order to reduce clutter we will drop the i subscript. For $1 \leq t \leq T$, define

$$Y_t = A_t + B_t X_t$$

where X_t is a scalar, and A_t, B_t are scalar random coefficients that evolve according to an AR(1) Markov process.

$$A_t = A_{t-1} + U_t \tag{5}$$

$$B_t = B_{t-1} + V_t \tag{6}$$

We will make the following sufficient assumptions for identifiability. Some of these are known to be stronger than necessary, and we will mention how to relax them later on. For ease of notation, let us denote $W_t = [X_t, X_{t-1}]^T$

Assumption 3.1. *Conditional on recent covariates W_t , contemporaneous shocks are independent from past shocks.*

$$\text{For } 2 \leq t \leq T, \quad (U_t, V_t) \perp \{U_s, V_s\}_{s=0}^{t-1} \mid W_t \quad (7)$$

Assumption 3.2. *Shocks are sequentially independent from all covariates*

$$\forall t, \quad U_t, V_t \perp (X_1, \dots, X_t) \quad (8)$$

Assumption 3.3. *The covariate associated with the random coefficient is continuously distributed and is supported on the entire real line for all periods. Furthermore, their joint distribution is non-degenerate.*

$$\text{supp}(X_1) = \dots = \text{supp}(X_T) = \mathbb{R} \quad \text{and} \quad \forall s \neq t, \quad P(X_s = X_t) = 0 \quad (9)$$

Assumption 3.4. *Nonsingularity of the covariate matrix.*

$$\text{For } 2 \leq t \leq T, \quad E[W_t W_t^T \mid X_2 - X_1 = 0] \quad \text{is nonsingular} \quad (10)$$

We should note that assumptions 3.1 and 4.5 allow for future regressors to be correlated with past shocks. In the context of firm profit maximization, this means that a firm is allowed to choose future inputs based on their previous productivity. In a different context, this model would also allow for a dynamic model with $X_t = Y_{t-1}$. Crucially, we do not make any assumptions of contemporaneous independence between random coefficients and regressors.

Assumption 3.3 is a simplifying “no-stayer” assumption that will facilitate some of the asymptotic derivations below. Assumption 3.4 is a common technical assumption often used for generic identification in linear models.

4 Identifying random coefficient moments

In this section, we show how to identify the r^{th} moments of the random coefficients (A_t, B_t) . Broadly, our identification strategy will be constructive, and follow these four steps.

1. Identify shock moments $E[U_t^\ell V_t^{r-\ell}]$
2. Express the conditional moments of the regressand $E[Y_1^\ell Y_2^{r-\ell} | W_{2i}]$ as a function of conditional moments of random coefficients $E[A_t^\ell B_t^{r-\ell} | W_2]$ and shocks $E[U_t^\ell V_t^{r-\ell}]$
3. Invert the above so as to represent conditional moments of random coefficients as a function of conditional moments of the regressand and shocks $E[U_t^\ell V_t^{r-\ell} | W_2]$
4. Integrate out to get unconditional moments $E[A_t^\ell B_t^{r-\ell}]$

4.1 First period moments

Let us start with the first moment ($r = 1$). In this case, we are required to have $T \geq 2$ consecutive waves of panel data.

We begin by simply reminding ourselves of the system of equations for these two periods.

$$\begin{cases} Y_1 = A_1 + B_1 X_1 \\ Y_2 = A_2 + B_2 X_2 \end{cases} \quad (11)$$

Let's expand the coefficients according to our model equations 3.

$$\begin{cases} Y_1 = A_1 + B_1 X_1 \\ Y_2 = A_1 + U_2 + B_1 X_2 + V_2 X_2 \end{cases} \quad (12)$$

Step 1 In order to identify shocks, we consider the expectation of the difference between the two periods, conditioning on some point $W_2 = w_2 = (x, x)$.

$$E[Y_2 - Y_1 \mid W_2 = w_2] = E[U_2] + E[V_2]x \quad (13)$$

The expression above contains two caveats. First, that thanks our full support assumption 3.3, we can choose any point on \mathbb{R}^2 to condition on, including a point where $X_1 = X_2 = x$ as we just did. Second, that Assumption 3.1 allowed us to drop the conditioning for the shocks.

Now, note that the moments $E[U_2]$ and $E[V_2]$ on the right-hand side are solutions to this minimization problem

$$\min_{\theta_u, \theta_v} E[(Y_2 - Y_1 - \theta_u - x\theta_v)^2 \mid W_2 = w_2] \quad (14)$$

and that the solution is unique, by virtue of the strict convexity of the quadratic function and the full rank assumption. It follows that the shock moments $E[U_2]$ and $E[V_2]$ are generically identified, and so are β_1 and β_2 .

Step 2 For this step, we simply have to take expectations of 12, now conditioning on $W_2 = w_2 = (x_1, x_2)$. (Note we have dropped the constraint $X_1 = X_2$).

$$\begin{cases} E[Y_1 \mid W_2] = E[A_1 \mid W_2] + E[B_1 \mid W_2]x_1 \\ E[Y_2 \mid W_2] = E[A_1 \mid W_2] + E[U_2] + E[B_1 \mid W_2]x_2 + E[V_2]x_2 \end{cases} \quad (15)$$

Step 3 Writing the previous equation in matrices after a little rearranging, we get:

$$\begin{bmatrix} E[Y_1|W_2] \\ E[Y_2|W_2] - E[U_2] - E[V_2]x_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} E[A_1|W_2] \\ E[B_1|W_2] \end{bmatrix} \quad (16)$$

As long as $x_1 \neq x_2$, we can invert the first matrix on the right-hand side.

$$\begin{bmatrix} E[A_1|W_2] \\ E[B_1|W_2] \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix}^{-1} \begin{bmatrix} E[Y_1|W_2] \\ E[Y_2|W_2] - E[U_2] - E[V_2]x_2 \end{bmatrix} \quad (17)$$

since all the quantities on the right-hand side are known, we have identified all the conditional first moment of our random coefficients outside the line $X_1 = X_2$. Since our “no-stayer” assumption 3.3 guarantees that a randomly drawn point from the joint distribution of X_1 and X_2 will not be on the diagonal, we have generic identification.

Conditional second-period moments are also automatically identified, since due to our 4.5 assumption we have that shocks are mean-independent from contemporaneous regressors, which allows us to write

$$E[A_2|W_2 = w_2] = E[A_1|W_2 = w_2] + E[U_2] \quad (18)$$

$$E[B_2|W_2 = w_2] = E[B_1|W_2 = w_2] + E[V_2] \quad (19)$$

$$(20)$$

Step 4 The final step is simply to integrate out over the distribution of W_2 to get unconditional moments.

4.2 Second period moments

The identification procedure is analogous, but this time we use the square of our model equations.

Step 1 Identify shock second moments on the diagonal $W_2 = w_2 = (x, x)$ using the following equation

$$E[(Y_2 - Y_1)^2 | W_2 = w_2] = E[U_2^2] + E[V_2^2]x^2 + 2E[U_2V_2]x \quad (21)$$

The analogous minimization procedure is

$$E[U_2^2], E[V_2^2], E[U_2V_2] = \min_{\theta_{uu}, \theta_{vv}, \theta_{uv}} E[(Y_2 - Y_1 - \theta_{uu} - x^2\theta_{vv} - 2x\theta_{uv})^2 | W_2 = w_2] \quad (22)$$

point identification is once more possible because the solution to this minimization problem is unique due to the rank assumption and the fact that the squaring function is strictly convex.

Steps 2-3 Again drop the constraint $X_1 = X_2$, and conditioning on $W_2 = w_2 = (x_1, x_2)$ consider the expectation of the square of our model equations.

$$\begin{bmatrix} E[A_1^2 | x_1, x_2] \\ E[B_1^2 | x_1, x_2] \\ E[A_1B_1 | x_1, x_2] \end{bmatrix} = \begin{bmatrix} 1 & x_1^2 & 2x_1 \\ 1 & x_2^2 & 2x_2 \\ 1 & x_1x_2 & x_1 + x_2 \end{bmatrix}^{-1} \begin{bmatrix} E[Y_1^2 | x_1, x_2] \\ E[Y_2^2 | x_1, x_2] - C_{1i} \\ E[Y_1Y_2 | x_1, x_2] - C_{2i} \end{bmatrix} \quad (23)$$

where C_{1i} and C_{2i} involve only quantities that were already estimated in previous

steps.

$$C_{1i} = E[U^2] + 2E[U_2V_2]x_2 + E[V_2^2]x_2^2 \quad (24)$$

$$- 2\{E[A_1|x_1, x_2] + E[B_1|x_1, x_2]x_2\}\{E[U_2] + E[V_2]x_2\} \quad (25)$$

$$C_{2i} = \{E[A_1|x_1, x_2] + E[B_1|x_1, x_2]x_1\}\{E[U_2] + E[V_2]x_2\} \quad (26)$$

Note that the relevant matrix above is invertible whenever $x_1 \neq x_2$.

Step 4 Analogous to step 4 in the first moment case.

4.3 Identifying further moments

The argument above generalizes for arbitrary moments. In fact, if we are only after moments a finite number of moments of the distribution, then assumption 4.5 as follows. Let r be a positive integer and for $t \geq 2$ define $W_t = (X_{t-1}, X_t)$. We will say that (U_t, V_t) are r^{th} -moment independent of W_t if

Definition 4.1.

$$\text{For } 1 \leq j \leq r, \quad E[U_t^{r-j}V_t^j|W_t] = E[U_t^{r-j}V_t^j] \quad (27)$$

We can replace assumption 4.5 by the following weaker assumption.

Assumption 3.1' *Shocks are r^{th} -moment independent from all covariates*

$$\text{For } 1 \leq j \leq r, \quad E[U_t^{r-j}V_t^j|W_t] = E[U_t^{r-j}V_t^j] \quad (28)$$

4.4 Extension: fixed coefficients

Our model can be augmented with an arbitrary number of regressors associated with coefficients that are individual-invariant – but not necessarily time-invariant. The model equations become

$$Y_t = A_t + B_t X_t + Z_t^T \beta_t \quad \beta_t : \text{fixed}$$

Identification of the fixed coefficients happens on Step 1, when the minimization problem becomes

$$E[U_2], E[V_2], \beta_1, \beta_2 = \min_{\theta_u, \theta_v, \theta_{b_1}, \theta_{b_2}} E[(Y_2 - Y_1 - a - bx_2 + z_2^T \theta_{b_1} - z_1^T \theta_{b_2})^2 | W_2 = w_2] \quad (29)$$

and the argument follows for the same reasons that were previously explained. Once we have identified β_1, β_2 , we remove them from future computations using the modified regressand $\tilde{Y}_t := Y_t - Z_t \beta_t$, and the rest of the process delineated above goes through without further change.

4.5 Identifying the marginal distribution of random coefficients

In this section we will prove that we can identify the marginal density of random coefficients $f_{A_t B_t}$. We begin by strengthening some of the assumption above.

Assumption 1.1' *Shocks are jointly independent of past shocks and current and past regressors*

$$\text{For } 1 \leq j \leq r, \quad (U_t, V_t) \perp (U_1, V_1, \dots, U_{t-1}, V_{t-1}, X_1, \dots, X_{t-1}) \quad (30)$$

Assumption 1.5 *The joint distribution of shocks is positive everywhere*

$$\text{For } 1 \leq j \leq r, \quad \phi_{U_t, V_t}(s_1, s_2) := E[\exp(i[s_1 U_{t_1} + s_2 V_{t_2}])] > 0 \quad \forall t \in \mathbb{R}^2 \quad (31)$$

Theorem 1 (Nonparametric identification). *If Assumptions 1.1', 1.3, 1.4 and 1.5 are satisfied, then the joint distribution of random coefficients f_{A_t, B_t} is identified for all t .*

Proof sketch Consider two adjacent periods, here denoted by $t = 1, 2$. Let $\phi_{Y_1, Y_2}(s_1, s_2 | x_1, x_2) := E[\exp(i[s_1 Y_1 + s_2 Y_2]) | X_1 = x_1, X_2 = x_2]$ be the joint characteristic function of the dependent variable conditional on observable covariates. We can identify the characteristic function of shocks by evaluating this characteristic function at the “diagonal” $X_1 = X_2 = x$ with $s_2 = -s_1 =: s$.

$$\phi_{Y_1, Y_2}(-s, s | x, x) = E[\exp(i[s Y_2 - s Y_1]) | X_1 = X_2 = x] \quad (32)$$

$$= E[\exp(i[s U_2 - s x V_2]) | X_1 = X_2 = x] \quad (33)$$

$$= E[\exp(i[s U_2 - s x V_2])] \quad \because \text{Assumption 1.1'} \quad (34)$$

$$=: \phi_{U_2, V_2}(s, s') \quad s' := s x \quad (35)$$

Applying the inverse Fourier transform to the joint characteristic function of the shocks ϕ_{U_2, V_2} , we retrieve their distribution f_{U_2, V_2} . Note that the full support Assumption 1.3 was used in the inversion step.

Next, as in the moment identification discussion, write

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \Gamma_1(x_1, x_2) \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} + \Gamma_2(x_1, x_2) \begin{bmatrix} U_2 \\ V_2 \end{bmatrix} \quad (36)$$

for appropriate matrices Γ_1 and Γ_2 . Now choose $t_1 = s^T \Gamma_1$ and $t_2 = s^T \Gamma_1^{-1}(x_1, x_2) \Gamma_2(x_1, x_2)$.

Note that this exists whenever $x_1 \neq x_2$. The conditional characteristic function of Y_1, Y_2 then decomposes multiplicatively:

$$\phi_{Y_1, Y_2}(s_1, s_2 | x_1, x_2) = E[\exp(is_1 Y_1 + is_2 Y_2) | X_1 = x_1, X_2 = x_2] \quad (37)$$

$$= E[\exp(it_1 [A_1, B_1]^T + it_2^T [U_2, V_2]^T) | X_1 = x_1, X_2 = x_2] \quad (38)$$

$$= E[\exp(it_1 [A_1, B_1]^T | X_1 = x_1, X_2 = x_2) \cdot E[\exp(it_2 [U_2, V_2]^T)]] \quad (39)$$

$$=: \phi_{A_1, B_1}(t_1, t_2 | x_1, x_2) \cdot \phi_{U_2, V_2}(t_2) \quad (40)$$

Since we have already identified $\phi_{U_2, V_2}(s)$ for all $s \in \mathbb{R}^2$, we can retrieve $\phi_{Y_1, Y_2}(t_2 | x_1, x_2)$ by dividing both sides of Equation 40 by $\phi_{U_2, V_2}(t_2)$ whenever t_2 exists. This implies that we can retrieve the conditional density function $f_{A_1, B_1}(a_1, b_1 | x_1, x_2)$ by applying the inverse Fourier transform to the ratio $\frac{\phi_{Y_1, Y_2}(t_2 | x_1, x_2)}{\phi_{U_2, V_2}(t_2)}$. Finally, averaging over X_1, X_2 , we can retrieve the marginal density f_{A_1, B_1} .²

5 Estimation

Our estimation procedure follows three steps:

1. Use local polynomial regression to estimate shock moments and individual-invariant coefficients
2. Use any nonparametric estimator to retrieve estimates of random coefficient moments conditional on observables
3. Average out conditional random moments by averaging out conditional moments, taking care to avoid observations for which $X_1 \approx X_2$.

Let us consider each one of these steps in detail.

²We are glossing over the fact that we have not identified the characteristic function over the diagonal $x_1 = x_2$. For the complete argument please see a supplementary note.

Estimating shocks

First moments Regress ΔY_2 on X_2 locally at the diagonal $X_1 \approx X_2$:

$$(\widehat{E}[U_2], \widehat{E}[V_2], \cdot) = \arg \min_{\theta_U, \theta_V, \gamma} \sum_i K_h(\Delta X_{2i}) \cdot (\Delta Y_{2i} - \theta_U - X_{2i}\theta_V - g_1(X_{2i}, \Delta X_{2i}; \gamma))^2 \quad (41)$$

where $K_h(\cdot)$ is a standard kernel with asymptotically zero bandwidth:

$$h_{shocks}(n) = c_{shocks} n^{\alpha_{shocks}} \quad \text{where } \lim_{n \rightarrow \infty} h_n = 0 \quad (42)$$

and g_1 is a K -order polynomial in X_2 and ΔX_2 with coefficients γ . In theory its inclusion is optional, but we have observed that its presence substantially improves the quality of the estimates.

If our model included time-invariant coefficients, they would also be estimated in this step.

Second moments We estimate the uncentered moments by proceeding similarly to above:

$$(\widehat{E}[U_2^2], \widehat{E}[U_2 V_2], \widehat{E}[V_2^2], \cdot) = \quad (43)$$

$$\arg \min_{\theta_{U_2}, \theta_{UV}, \theta_{V_2}, \gamma} \sum_i K_h(\Delta X_{2i}) \cdot (\Delta Y_{2i}^2 - \theta_{U_2} - 2X_{2i}\theta_{UV} - X_{2i}^2\theta_{V_2} - g_2(X_{2i}, \Delta X_{2i}; \gamma))^2 \quad (44)$$

Naturally, once we have uncentered second moments and first moments, we can compute estimates of centered moments using the usual formulas for centered moments, e.g. $\widehat{Var}[U_2] = \widehat{E}[U_2^2] - \widehat{E}[U_2]^2$.

Conditional random coefficient moments

Conditional moments of Y_1 and Y_2 Here we begin by estimating $E[Y_1^m Y_2^n | X_1 = x_1, X_2 = x_2]$ for $m, n \in \{(1, 0), (0, 1), (1, 1), (2, 0), (0, 2)\}$. This can be done using any nonparametric estimator, but we choose the Nadaraya-Watson for simplicity.

$$\hat{E}[Y_1 | X_1 = x_1, X_2 = x_2] = \frac{\sum_i Y_{1i} K_h(X_{1i} - x_1) K_h(X_{2i} - x_2)}{\sum_i K_h(X_{1i} - x_1) K_h(X_{2i} - x_2)} \quad (45)$$

where K_h is again a standard kernel endowed with asymptotically vanishing bandwidth

$$h_{nw}(n) = c_{nw} \hat{\sigma}_X n^{-\alpha_{nw}} \quad (46)$$

and $\hat{\sigma}_X$ is an estimate of $Std(X_1) = Std(X_2)$. Note that for the theoretical MISE-minimizing bandwidth, one should set $\alpha_{nw} = \frac{1}{6}$.

Solving for first moments Once in possession of all the previous estimates, we can solve for estimates of conditional random coefficient moments using the empirical analogs of equation 4.1.

Unconditional random coefficient moments

In this final step, we must exclude observations near the diagonal $X_1 X_2$, and then average the conditional moments associated with the remaining observations. For the first moment of the intercept, we compute

$$\hat{E}[A_1] = \frac{\sum_{i=1}^n \hat{E}[A_1 | X_{1i}, X_{2i}] \cdot 1\{|X_{2i} - X_{2i}| > t_1(n)\}}{\sum_{i=1}^n 1\{|X_{2i} - X_{2i}| > t_1(n)\}} \quad (47)$$

and analogously for the slope and for higher moments. Here, $t_1(n)$ and $t_2(n)$ (for

second moments) are *censoring threshold* functions that vanish asymptotically. Consistency and asymptotic normality of the resulting estimates are ensured provided that

$$t_1(n) \equiv c_{cens1} \hat{\sigma}_{X_2} n^{-\frac{1}{4}} \quad \text{for first moments} \quad (48)$$

$$t_2(n) \equiv c_{cens1} \hat{\sigma}_{X_2} n^{-\frac{1}{8}} \quad \text{for second moments} \quad (49)$$

where c_{cens1} , c_{cens2} are constants that are not pinned down by theory. In order to choose these constants, we resort to experimentation as shown in section 7.

6 Asymptotic results

In this section we present an abridged version of our asymptotic results without proof.³

Definition 6.1. Define the following objective functions

$$\hat{G}_n(\alpha) = \frac{1}{n} \sum_{i=1}^n \left[\left(\hat{E}(A_1 | X_{1i}, X_{2i}) - \alpha \right)^2 1\{|X_{2i} - X_{1i}| > h_n\} \right] \quad (50)$$

$$\Gamma(\alpha) = E \left[\left(\hat{E}(A_1 | X_1, X_2) - \alpha \right)^2 \right] \quad (51)$$

and let $\hat{\alpha}$ and α_0 be the minimizers of 50 and 51. Explicitly:

$$\hat{\alpha} = \frac{\sum_{i=1}^n \hat{E}[A_1 | X_{1i}, X_{2i}] 1\{|X_{2i} - X_{1i}| > h_n\}}{\sum_{i=1}^n 1\{|X_{2i} - X_{1i}| > h_n\}} \quad (52)$$

$$\alpha_0 = E[E[(A_1 | X_1, X_2)]] = E[A_1] \quad (53)$$

Theorem 2 (Consistency). *If the following two conditions hold*

1. $\sup_{\alpha} \left| \hat{G}_n(\alpha) - \Gamma(\alpha) \right| \in o_p(1)$

³For an elaborated discussion, please see supplementary material available upon request.)

2. For each $\delta > 0$, $\inf_{\alpha: |\alpha - \alpha_0| < \delta} \Gamma(\alpha) > \Gamma(\alpha_0)$

Then $|\hat{\alpha} - \alpha_0| \in o_p(1)$

Theorem 3 (Asymptotic Normality for conditional moments). *Suppose that $x_1 \neq x_2$ and $f(x_1, x_2) > 0$. If the following conditions hold*

1. For $k \in \{0, 1\}$, the function $r_k(x_1, x_2, \theta)$ defined below has continuous second partial derivative with respect to x_1 .

$$r_k(x_1, x_2, \theta) = E[(Y_2 - Y_1 - EU_2 - EV_2X_2)X_2^k | X_1 = x_1, X_2 = x_2]f(X_1 = x_1, X_2 = x_2) \quad (54)$$

2. The function $s_t(x_1, x_2) = E[Y_1 | X_1 = x_1, X_2 = x_2]f(X_2 = x_2 | X_1 = x_1)$ is twice continuously differentiable

3. The marginal density of the first regressor $f(x_1)$ is twice continuously differentiable

4. The conditional means of Y_1 and Y_2 are estimated with uniform kernels endowed with bandwidth $h_{nw}(n) \propto n^{-\alpha_{nw}}$, where $\frac{1}{5} < \alpha_{nw} < \frac{1}{2}$

5. The censoring threshold functions decay as the following rates $t_1(n) \propto n^{-\frac{1}{4}}$ and $t_2 \propto n^{-\frac{1}{8}}$

Then the estimates of conditional first and second moments are asymptotically normal.

7 Simulations

The purpose of this sections is to provide a simulation study in a setting that is similar to our empirical application. We begin by generating the first period random coeffi-

cients (A_1, B_1) , their second-period shocks (U_2, V_2) , and regressors for both periods (X_1, X_2) . They are drawn from a jointly Normal distribution:

$$\begin{bmatrix} A_1 \\ B_1 \\ X_1 \\ X_2 \\ U_2 \\ V_2 \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 2 \\ .4 \\ 0 \\ 0 \\ .3 \\ .1 \end{bmatrix}, \begin{bmatrix} 9 & 0.95 & 1.5 & 1.5 & 0 & 0 \\ 0.95 & 0.4 & 0.32 & 0.32 & 0 & 0 \\ 1.5 & 0.32 & 1 & 0.5 & 0 & 0 \\ 1.5 & 0.32 & 0.50 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.16 \\ 0 & 0 & 0 & 0 & 0.16 & 0.1 \end{bmatrix} \right) \quad (55)$$

These number were chosen to roughly reflect the characteristics that we expect them to have in a real data application. The implied correlation matrix has a very simple structure.

$$\begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 1 \end{bmatrix} \quad (56)$$

The remaining variables were created from these in the obvious manner. In order to understand how our estimator behaves as the number of observation increases, we used $n \in \{500, 2000, 5000, 10000, 20000\}$.

As seen in Section 5, our method requires the careful tuning of many different parameters. Since our methods are not amenable to cross-validation, we simulated the model in section 1 over a large grid of parameter combinations, which were then

compared according to their RMSE against the true value. The RMSE-minimizing parameter configuration for each number of observations is shown on Table 14.

n	C_{shocks}	α_{shocks}	C_{nw}	α_{nw}
500	2	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{6}$
2000	3	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{6}$
5000	4	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{6}$
10000	4	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{6}$
20000	3	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{3}$

Table 14: RMSE-minimizing parameter configurations for our DGP.

Note that the best parameters remain fairly stable as we increase the number of observations. The most notable variation is in α_{nw} : intuitively, when the number of observations is large enough, a much smaller bandwidth is preferred, and that is translated into a more aggressive choice of exponent. We will use these numbers to guide our choice of tuning parameters in our empirical application in the next section.

The performance of our estimators is illustrated on Figures 13 and 14 (see also accompanying tables in the Appendix). We observe that indeed the distribution tends to concentrate around its true value as the number of observations increases.

7.1 Bootstrap coverage

Tables 15-17 show the coverage of bootstrap confidence intervals for all estimated parameters. To produce these tables, we generated 2000 datasets using different seeds, computed 500 bootstrap estimates in each of these datasets, and calculated their 0.025 and 0.975 quantiles. The numbers in each cell are the proportion of times that quantiles straddled the true value.

Table 15: Bootstrap coverage for shock moments

	$E[U_t]$	$E[V_t]$	$Std[U_t]$	$Std[V_t]$	$Cov[U_t, V_t]$	$Corr[U_t, V_t]$
t=1	0.948	0.942	0.952	0.968	0.942	0.950
t=2	0.951	0.941	0.945	0.964	0.941	0.954

Table 16: Bootstrap coverage for random coefficient conditional moments

(x_1, x_2)	$(-1, 0)$	$(1, 0)$	$(0, 1)$	$(0, -1)$
$E[A_1 X]$	0.960	0.920	0.957	0.919
$E[B_1 X]$	0.936	0.929	0.923	0.953
$Std[A_1 X]$	0.920	0.950	0.945	0.932
$Std[B_1 X]$	0.952	0.929	0.932	0.953
$Cov[A_1, B_1 X]$	0.940	0.958	0.928	0.936
$Corr[A_1, B_1 X]$	0.932	0.966	0.932	0.940

Table 17: Bootstrap coverage for random coefficient unconditional moments

	$E[A_t]$	$E[B_t]$	$Std[A_t]$	$Std[B_t]$	$Cov[A_t, B_t]$	$Corr[A_t, B_t]$
t=1	0.945	0.958	0.951	0.928	0.948	0.945
t=2	0.935	0.954	0.951	0.932	0.945	0.941

8 Empirical application

Now we provide an illustration of our methods by estimating a Cobb-Douglas production function with random coefficients for a large number of Indian plants. We will be using the Annual Survey of Industries (ASI) dataset, released by the Central Statistical Organization of India for the years 2008 and 2009. This yearly survey collects data from sampled Indian economic units of production (individual factories, workshops, and establishments, hereafter called “plants”) that employ more than 10 regular workers.

From the ASI dataset we extract four variables: *gross sale value* (value of the products sold by the plant, as purchased by their clients) S_t ; *capital* (fixed assets with a productive life of more than one year) K_t ; *wages* W_t ; and production *materials* M_t . All variables are in 2005 rupees. We keep only observations that we present in both years of analysis, and we are left with 13298 observations.⁴

In addition to the variables above, we generate our model variables:

$$Y_t = \log\left(\frac{S_t - M_t}{W_t}\right) \quad X_t = \log\left(\frac{K_t}{W_t}\right) \quad (57)$$

where Y_t represents production value-added after sales normalized by wages, and X_t represents normalized capital. Summary statistics for all variance can be found in subsection 10.1 in the Appendix.

Following Allcott et al. (2016), we also construct variables `nic1987`, corresponding to the 1987 Indian three-digit National Industry Code, `state` indicating the plant location, and `panel_group` that is a concatenation of the state code and the first two digits of `nic1987`.

Interpretation Each plant's output is modeled as a Hicks-neutral Cobb-Douglas function. Therefore, the random variable A_t represents *total factor productivity*, while B_t represents the expenditure share on capital inputs.

Included in the modeling decision above there are several assumptions. First, timing: we are assuming that each firm observes the entire history of past inputs and outputs $(Y_1, X_1, \dots, Y_{t-1}, X_{t-1})$, as well as the values of their own TFP and capital share $(A_1, B_1, \dots, A_{t-1}, B_{t-1})$. Based on this information, they choose this period's inputs X_t . Once the decision is taken, this period's shocks U_t, V_t are revealed, and production Y_t is realized.

⁴Additional information about data cleaning and manipulation is available as Jupyter Notebook at https://github.com/halflearned/FHHPS/blob/master/empirical/FHHPS_data_cleaning.html

Second, we have left the stochastic process governing the evolution of X_t unrestricted. In particular, we do not impose the assumption that firms are profit-maximizers. This is important for studying unproductive firms as well.

Discussion The result of our empirical application are presented on Tables 18-20. Additionally, we present the results of similar ordinary panel regressions and quantile panel regressions on Tables 37 and 38⁵ Our main findings are as follows.

First, our estimates of the first moment of random coefficients are quite similar to those given by traditional linear panel data models, with $\widehat{E}[A_1] \approx 1.5$ and $\widehat{E}[B_1] \approx 0.25$. This indicates that allowing only for scalar-valued additive heterogeneity would not preclude reasonably accurate estimation in this particular data set.

Second, our estimates suggest more heterogeneity than traditional models. The fixed-effect quantile regression result on Table 38 indicates that the 0.025- and 0.975-quantiles of the slope parameter are around 0.23 and 0.30, respectively. On the other hand, our results output $\widehat{Std}[B_1] \approx 0.57$, indicating that the support of the slope is much larger.⁶

Third, while the distribution of total factor productivity and capital share seems to vary a lot across plant, their average tends to be fairly stable over time, as we understand from $\widehat{E}[U_2] \approx \widehat{E}[V_2] \approx 0$. Fourth and finally, total factor productivity and capital share seem to be quite strongly negatively correlated, with $\widehat{Corr}[A_1, B_1] = -0.44$. We could not find a satisfactory explanation for this phenomenon, especially in light of the fact that we have at the same time $\widehat{Corr}[U_2, V_2] = 0.91$, and leave it for future research.

⁵Quantile panel regressions were run using the R package `rqpd` that is based in Koenker and Hallock (2001) and Koenker (2004).

⁶ Incidentally, our standard deviation estimate of $\widehat{Std}[B_1] = 0.57$ would also imply that $supp(B_1)$ is not a subset of the unit interval: by Popoviciu (1935)'s inequality, the maximum attainable variance for random variable Z with $supp(Z) = [0, 1]$ is $\frac{1}{4}$.

Table 18: Empirical application: Shock Estimates

	$E[U_2]$	$E[V_2]$	$Std[U_2]$	$Std[V_2]$	$Cov[U_2, V_2]$	$Corr[U_2, V_2]$
Estimate	0.011	-0.002	0.704	0.050	0.032	0.91
Std. Error	0.010	0.005	0.016	0.018	0.005	0.539
Min	-0.014	-0.013	0.664	0.008	0.018	0.463
Lower CI (2.5%)	-0.010	-0.012	0.67	0.014	0.023	0.572
Median	0.011	-0.001	0.702	0.052	0.032	0.872
Upper CI (97.5%)	0.028	0.007	0.73	0.086	0.044	2.612
Max	0.0303	0.010	0.735	0.095	0.047	3.696

Table 19: Empirical application: Random Coefficient Estimates (first period)

	$E[A_1]$	$E[B_1]$	$Std[A_1]$	$Std[B_1]$	$Cov[A_1, B_1]$	$Corr[A_1, B_1]$
Estimate	1.582	0.238	1.369	0.571	-0.342	-0.437
Std. Error	0.037	0.016	0.127	0.047	0.108	0.096
Min	1.470	0.193	1.050	0.452	-0.669	-0.668
Lower CI (2.5%)	1.513	0.205	1.121	0.477	-0.518	-0.615
Median	1.583	0.237	1.369	0.566	-0.339	-0.440
Upper CI (97.5%)	1.657	0.273	1.562	0.663	-0.142	-0.242
Max	1.687	0.286	1.602	0.697	-0.056	-0.088

Table 20: Empirical application: Random Coefficient Estimates (second period)

	$E[A_2]$	$E[B_2]$	$Std[A_2]$	$Std[B_2]$	$Cov[A_2, B_2]$	$Corr[A_2, B_2]$
Estimate	1.593	0.236	1.540	0.573	-0.310	-0.351
Std. Error	0.037	0.017	0.113	0.047	0.109	0.092
Min	1.468	0.194	1.274	0.452	-0.644	-0.544
Lower CI (2.5%)	1.521	0.206	1.318	0.479	-0.485	-0.514
Median	1.598	0.238	1.538	0.570	-0.308	-0.348
Upper CI (97.5%)	1.670	0.271	1.715	0.662	-0.112	-0.155
Max	1.694	0.289	1.751	0.699	-0.023	-0.001

Simulated shock moment estimates

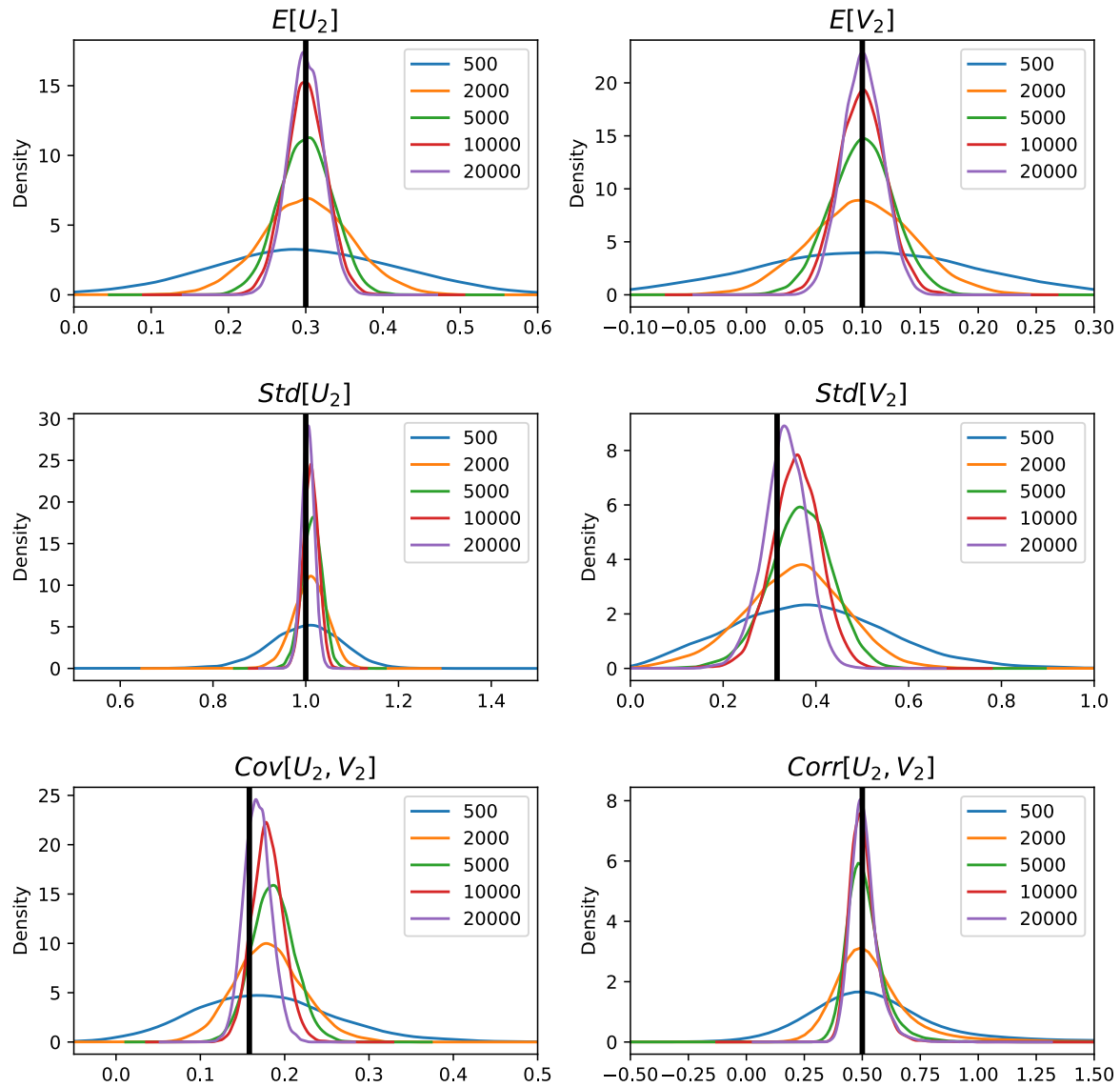


Figure 13: Shock estimates in simulated data

Simulated random coefficient moment estimates

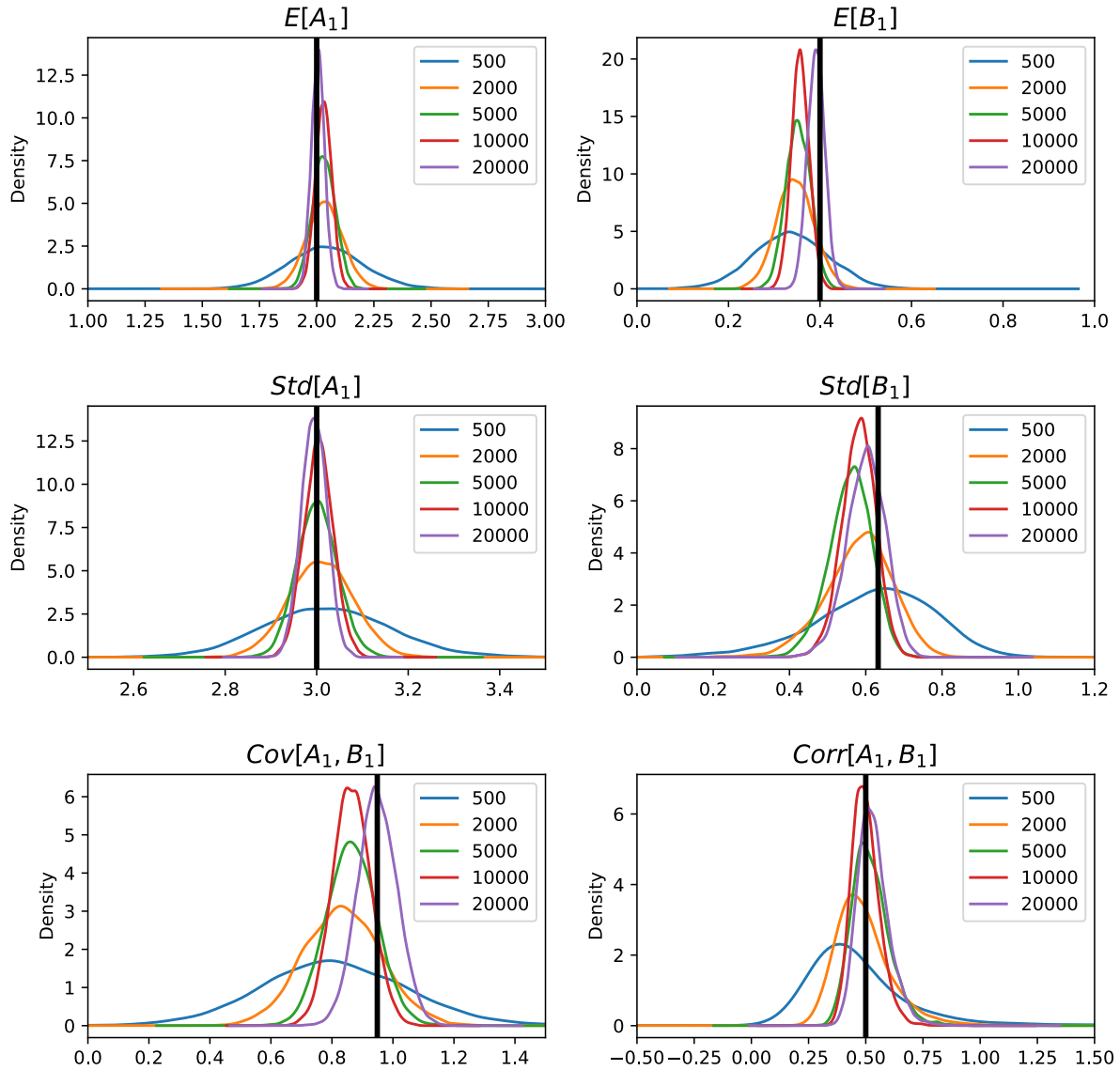


Figure 14: Random coefficient estimates in simulated data

Bootstrapped shock moments

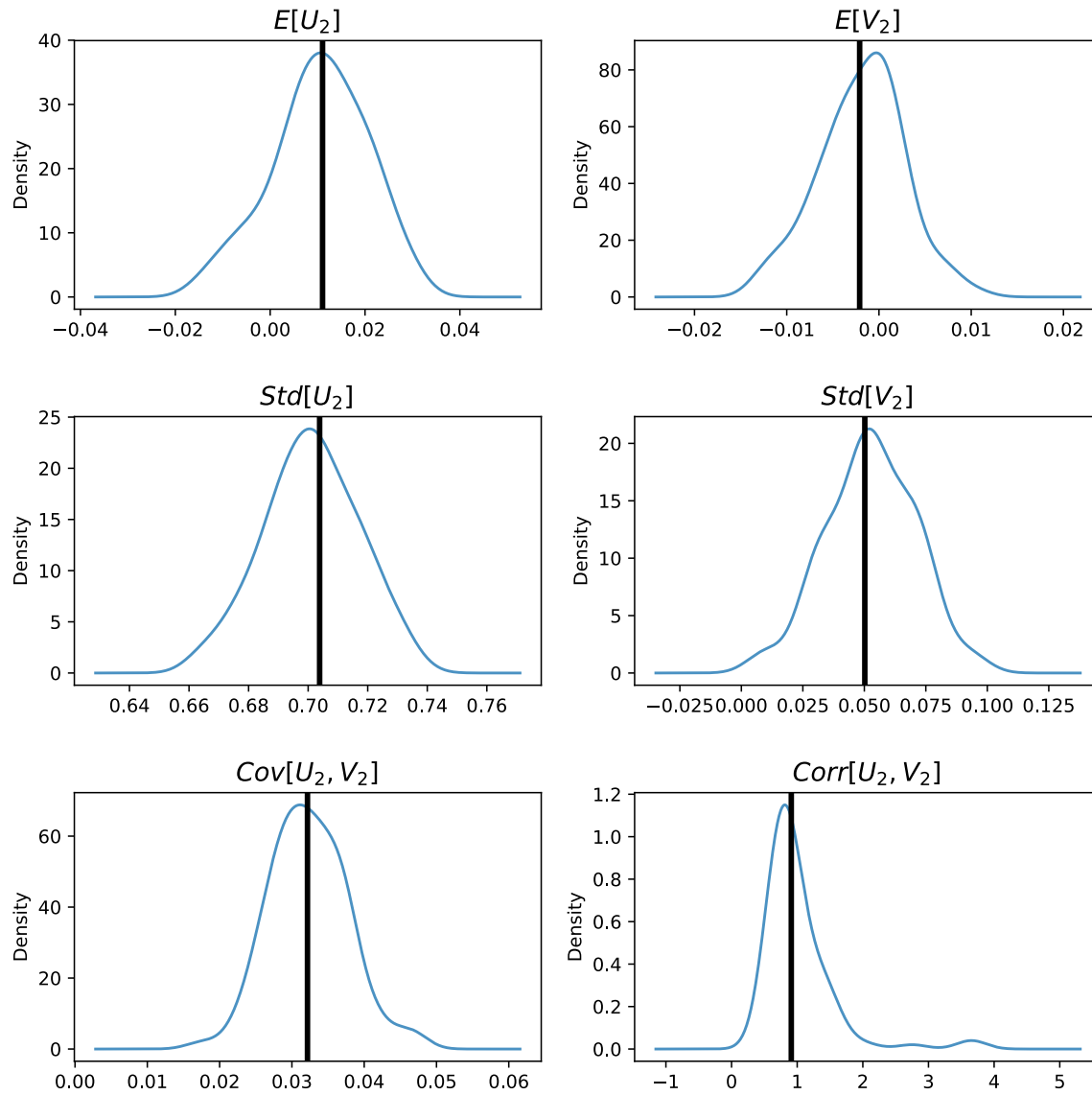


Figure 15: Shock estimates using ASI data

Bootstrapped random coefficient moments

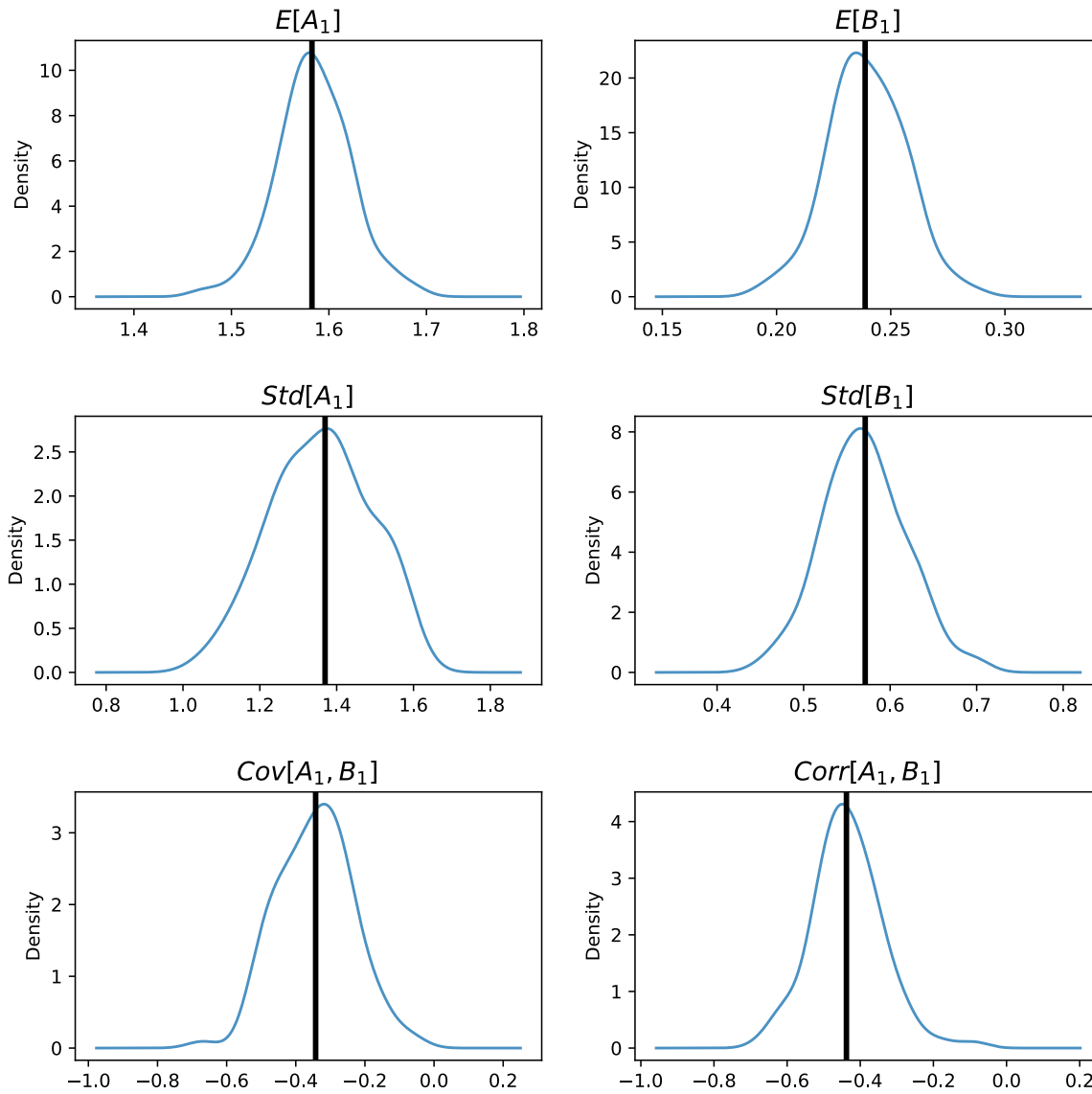


Figure 16: Random coefficient estimates using ASI data (2008)

Bootstrapped random coefficient moments

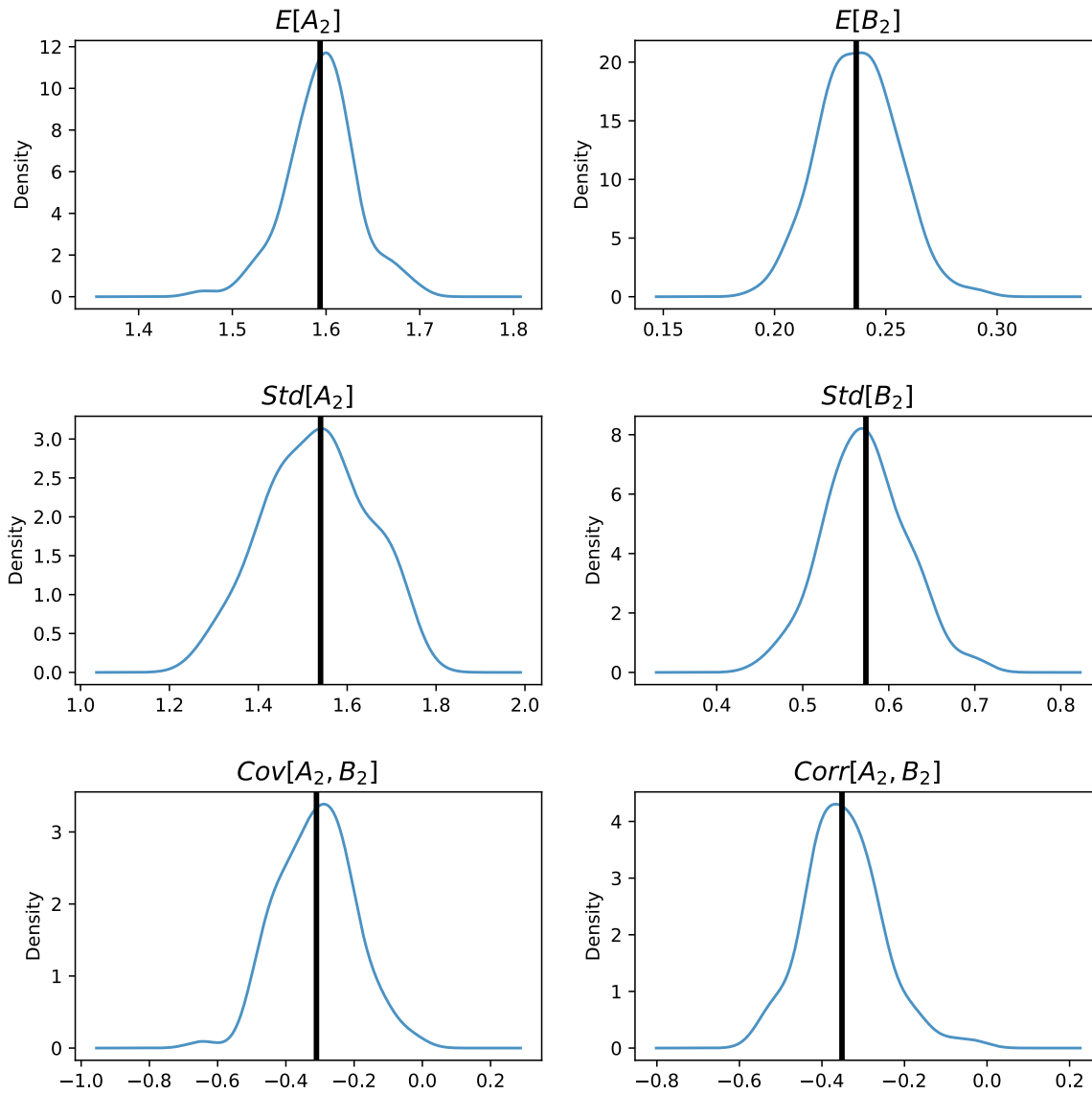


Figure 17: Random coefficient estimates using ASI data (2009)

9 Conclusion and future work

In this paper, we presented methods for identification of moments and nonparametric marginal distributions of endogenous random coefficient models in linear panel data models. Our identification strategy is constructive, immediately leading to a relatively simple estimators that are shown to be consistent and asymptotically normal, although they converge at rates slower than root- n due to irregular identification. We illustrate the use of our methods by estimating Cobb-Douglas coefficients in production functions on Indian plant-level microdata.

For future work, we leave the following research directions. First, our method requires the tuning of many parameters that are not amenable to cross-validation. Are there better heuristics than those we used here to deal with this issue? Second, our method allows a maximum of two correlated random coefficients, but one can conceive situations where this might be an appropriate assumption. Can we exploit a similar method and expand to more than two coefficients?

10 Appendix

10.1 Summary statistics

Table 21: ASI survey data: summary statistics

	K_{2008}	M_{2008}	W_{2008}	S_{2008}	K_{2009}	M_{2009}	W_{2009}	S_{2009}
mean	16.8	17.9	15.6	18.4	16.8	17.9	15.6	18.4
std	2.63	2.29	1.97	2.29	2.67	2.31	1.99	2.32
min	-0.137	8.3	8.14	8.1	-0.207	7.96	7.85	5.83
25%	15.0	16.3	14.1	16.7	15.0	16.3	14.1	16.8
50%	17.0	18.1	15.8	18.6	17.0	18.2	15.8	18.7
75%	18.7	19.5	17.1	20.0	18.7	19.6	17.1	20.1
max	26.3	27.1	23.3	27.4	26.2	27.2	23.2	27.4

Variables are capital K , materials M , wages W and sales value S . Variable units are 2005 rupees (in logs).

Table 22: ASI survey data: correlations

	K_{2008}	M_{2008}	W_{2008}	S_{2008}	K_{2009}	M_{2009}	W_{2009}	S_{2009}
K_{2008}	1.0	0.825	0.808	0.827	0.98	0.822	0.812	0.829
M_{2008}		1.0	0.843	0.976	0.823	0.963	0.834	0.948
W_{2008}			1.0	0.861	0.802	0.817	0.971	0.839
Y_{2008}				1.0	0.824	0.944	0.851	0.96
K_{2009}					1.0	0.828	0.814	0.834
M_{2009}						1.0	0.837	0.973
W_{2009}							1.0	0.859
Y_{2009}								1.0

Correlations between capital K , materials M , wages W and sales value S . Variable units are 2005 rupees (in logs).

Table 23: Transformed variables: summary statistics

	X_1	X_2	Y_1	Y_2
mean	1.05	1.09	1.83	1.82
std	1.51	1.51	1.09	1.11
min	-14.9	-15.9	-5.93	-7.12
25%	0.188	0.237	1.22	1.18
50%	1.15	1.18	1.83	1.82
75%	2.02	2.06	2.47	2.47
max	7.96	7.96	6.5	9.75

$X = \log\left(\frac{K}{W}\right)$ and $Y = \log\left(\frac{S-M}{W}\right)$

Table 24: Transformed variables: correlations

	X_1	X_2	Y_1	Y_2
X_1	1.0	0.911	0.43	0.401
X_2		1.0	0.394	0.433
Y_1			1.0	0.717
Y_2				1.0

$X = \log\left(\frac{K}{W}\right)$ and $Y = \log\left(\frac{S-M}{W}\right)$

10.2 Simulation results

Here we show how our estimates of shock and random coefficient moments change as we increase the number of observations. Throughout, we used the RMSE-minimizing parameters shown in table 14. To produce the tables below, we generated 8000 datasets for each number different number of observations, produced the relevant estimates and computed their respective performance measures.

Shock moments

Table 25: Performance measures for estimates of $E[U_2]$

n	bias	rel_bias	abs_bias	rmse
500	-0.0013	-0.0043	0.0979	0.1230
2000	0.0010	0.0032	0.0458	0.0576
5000	0.0000	0.0001	0.0275	0.0342
10000	-0.0001	-0.0002	0.0208	0.0262
20000	-0.0001	-0.0003	0.0181	0.0227

Table 26: Performance measures for estimates of $E[V_2]$

n	bias	rel_bias	abs_bias	rmse
500	0.0002	0.0023	0.0776	0.0972
2000	-0.0001	-0.0014	0.0356	0.0446
5000	0.0001	0.0006	0.0215	0.0270
10000	-0.0002	-0.0017	0.0163	0.0205
20000	0.0003	0.0029	0.0139	0.0173

Table 27: Performance measures for estimates of $\text{Std}[U_2]$

n	bias	rel_bias	abs_bias	rmse
500	-0.0021	-0.0021	0.0614	0.0780
2000	0.0096	0.0096	0.0302	0.0379
5000	0.0135	0.0135	0.0208	0.0256
10000	0.0109	0.0109	0.0159	0.0197
20000	0.0048	0.0048	0.0116	0.0145

Table 28: Performance measures for estimates of $\text{Std}[B_2]$

	bias	rel_bias	abs_bias	rmse
n				
500	-0.0120	-0.0189	0.1242	0.1586
2000	-0.0515	-0.0815	0.0793	0.1034
5000	-0.0733	-0.1159	0.0771	0.0923
10000	-0.0512	-0.0810	0.0559	0.0677
20000	-0.0324	-0.0512	0.0474	0.0601

Table 29: Performance measures for estimates of $\text{Cov}[U_2, V_2]$

	bias	rel_bias	abs_bias	rmse
n				
500	0.0186	0.1175	0.0681	0.0873
2000	0.0237	0.1499	0.0372	0.0473
5000	0.0282	0.1786	0.0312	0.0376
10000	0.0217	0.1375	0.0237	0.0285
20000	0.0097	0.0614	0.0147	0.0186

Table 30: Performance measures for estimates of $\text{Corr}[U_2, V_2]$

	bias	rel_bias	abs_bias	rmse
n				
500	0.1198	0.2397	0.2586	0.6747
2000	0.0736	0.1473	0.1387	0.3689
5000	0.0145	0.0290	0.0613	0.1130
10000	0.0050	0.0100	0.0456	0.0630
20000	0.0077	0.0154	0.0437	0.0601

Random coefficient moments

Table 31: Performance measures for estimates of $E[A_1]$

n	bias	rel_bias	abs_bias	rmse
500	0.0411	0.0205	0.1323	0.1669
2000	0.0344	0.0172	0.0683	0.0856
5000	0.0313	0.0157	0.0476	0.0591
10000	0.0300	0.0150	0.0381	0.0467
20000	0.0057	0.0029	0.0234	0.0293

Table 32: Performance measures for estimates of $E[B_1]$

n	bias	rel_bias	abs_bias	rmse
500	-0.0637	-0.1593	0.0843	0.1035
2000	-0.0540	-0.1349	0.0578	0.0677
5000	-0.0479	-0.1197	0.0486	0.0547
10000	-0.0433	-0.1083	0.0435	0.0474
20000	-0.0085	-0.0212	0.0164	0.0205

Table 33: Performance measures for estimates of $\text{Std}[A_1]$

n	bias	rel_bias	abs_bias	rmse
500	0.0229	0.0076	0.1119	0.1411
2000	0.0094	0.0031	0.0561	0.0702
5000	0.0013	0.0004	0.0352	0.0443
10000	0.0041	0.0014	0.0257	0.0323
20000	-0.0054	-0.0018	0.0226	0.0282

Table 34: Performance measures for estimates of $\text{Std}[B_1]$

	bias	rel_bias	abs_bias	rmse
n				
500	-0.0120	-0.0189	0.1242	0.1586
2000	-0.0515	-0.0815	0.0793	0.1034
5000	-0.0733	-0.1159	0.0771	0.0923
10000	-0.0512	-0.0810	0.0559	0.0677
20000	-0.0324	-0.0512	0.0474	0.0601

Table 35: Performance measures for estimates of $\text{Cov}[A_1, B_1]$

	bias	rel_bias	abs_bias	rmse
n				
500	-0.1401	-0.1477	0.2252	0.2790
2000	-0.1130	-0.1191	0.1396	0.1701
5000	-0.0840	-0.0885	0.0972	0.1174
10000	-0.0840	-0.0885	0.0892	0.1042
20000	0.0006	0.0006	0.0513	0.0643

Table 36: Performance measures for estimates of $\text{Corr}[A_1, B_1]$

	bias	rel_bias	abs_bias	rmse
n				
500	-0.0125	-0.0250	0.1913	0.3675
2000	-0.0042	-0.0085	0.0994	0.1554
5000	0.0220	0.0440	0.0652	0.0862
10000	-0.0010	-0.0019	0.0476	0.0607
20000	0.0333	0.0666	0.0576	0.0769

10.3 Python module FHHPS

In order to facilitate the adoption of the methods described in this paper by other researchers, I have developed the Python module **FHHPS**. Instructions for installations and usage can be found in the github repository ⁷. Its API is inspired by the celebrated **scikit-learn** library Buitinck et al. (2013), and it should be familiar to statisticians and machine learning practitioners that use Python. The webpage linked above also contains one-line instructions for reproducing all of our figures and tables.

⁷Url: <https://github.com/halfearned/FHHPS>

Table 37: Linear panel regressions

	$Y := \log\left(\frac{\text{Output-Materials}}{\text{Wages}}\right)$ and $X := \frac{\text{Capital}}{\text{Wages}}$					
	\tilde{Y} (F)	\tilde{Y} (P)	$\tilde{Y} + \text{factor}(\text{panel_group})$ (R)	\tilde{Y} (P)	$\tilde{Y} + \text{factor}(\text{state})$ (R)	$\tilde{Y} + \text{factor}(\text{nic1987})$ (R)
Intercept		1.489 (0.007)	1.495 (0.009)	1.361 (0.131)	1.360 (0.169)	1.318 (0.185)
X	0.277 (0.011)	0.315 (0.004)	0.309 (0.005)	0.272 (0.004)	0.272 (0.005)	0.292 (0.005)
R^2	0.045	0.186	0.135	0.271	0.196	0.202
Adjusted R^2	-0.909	0.186	0.135	0.262	0.186	0.201

Note: F: Fixed effects, R: Random effects, P: Pooling. All coefficients significant with $p < 0.01$. N=13298, T=2.

Table 38: Quantile panel regressions

Y ~X + year2009				
Quantile	Value	Std. Error	t value	<i>Pr(> t)</i>
X[0.025]	0.23587	0.01695	13.91812	0.00000
X[0.25]	0.26844	0.01408	19.07059	0.00000
X[0.5]	0.27424	0.01467	18.69515	0.00000
X[0.75]	0.28232	0.01433	19.69936	0.00000
X[0.975]	0.30993	0.01939	15.98118	0.00000

Bibliography

- Abdulkadiroğlu, A. and Sönmez, T. (1999). House allocation with existing tenants. *Journal of Economic Theory*, 88(2):233–260.
- Abraham, D. J., Blum, A., and Sandholm, T. (2007). Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 295–304. ACM.
- Agrawal, R. (1995). Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078.
- Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1.
- Akbarpour, M., Li, S., and Oveis Gharan, S. (2017). Thickness and information in dynamic matching markets.
- Allcott, H., Collard-Wexler, A., and O’Connell, S. D. (2016). How do electricity shortages affect industry? evidence from india. *American Economic Review*, 106(3):587–624.
- Anderson, R., Ashlagi, I., Gamarnik, D., and Roth, A. E. (2015). Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112(3):663–668.
- Arellano, M. (2003). *Panel data econometrics*. Oxford university press.
- Arellano, M. and Bond, S. (1991). Some tests of specification for panel data: Monte carlo evidence and an application to employment equations. *The review of economic studies*, 58(2):277–297.
- Arellano, M. and Bonhomme, S. (2011). Identifying distributional characteristics in random coefficients panel data models. *The Review of Economic Studies*, 79(3):987–1020.
- Ashlagi, I., Gamarnik, D., Rees, M. A., and Roth, A. E. (2012). The need for (long) chains in kidney exchange. Technical report, National Bureau of Economic Research.
- Ashlagi, I., Gilchrist, D. S., Roth, A. E., and Rees, M. A. (2011). Nonsimultaneous chains and dominos in kidney-paired donation revisited. *American Journal of transplantation*, 11(5):984–994.
- Ashlagi, I., Jaillet, P., and Manshadi, V. H. (2013). Kidney exchange in dynamic sparse heterogenous pools. *arXiv preprint arXiv:1301.3509*.

- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256.
- Awasthi, P. and Sandholm, T. (2009). Online stochastic optimization in the large: Application to kidney exchange. In *IJCAI*, volume 9, pages 405–411.
- Blundell, R. and Powell, J. L. (2003). Endogeneity in nonparametric and semiparametric regression models. *Econometric society monographs*, 36:312–357.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Browning, M., Carro, J., et al. (2007). Heterogeneity and microeconometrics modeling. *Econometric Society Monographs*, 43:47.
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.
- Cecka, J. (2010). Calculated pra (cpa): the new measure of sensitization for transplant candidates. *American journal of transplantation*, 10(1):26–29.
- Chamberlain, G. (1984). Panel data. *Handbook of econometrics*, 2:1247–1318.
- Chamberlain, G. (1992). Efficiency bounds for semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 567–596.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Cunha, F., Heckman, J., and Navarro, S. (2005). Separating uncertainty from heterogeneity in life cycle earnings. *oxford Economic papers*, 57(2):191–261.
- Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852.
- Dickerson, J. P., Kazachkov, A. M., Procaccia, A. D., and Sandholm, T. (2017). Small representations of big kidney exchange graphs. In *AAAI*, pages 487–493.
- Dickerson, J. P., Manlove, D. F., Plaut, B., Sandholm, T., and Trimble, J. (2016). Position-indexed formulations for kidney exchange. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 25–42. ACM.
- Dickerson, J. P., Procaccia, A. D., and Sandholm, T. (2012). Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI*.

- Dickerson, J. P. and Sandholm, T. (2015). Futurematch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI*, pages 622–628.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Gandhi, A., Navarro, S., and Rivers, D. A. (2011). On the identification of production functions: How heterogeneous is productivity?
- Graham, B. S. and Powell, J. L. (2012). Identification and estimation of average partial effects in irregular correlated random coefficient panel data models. *Econometrica*, 80(5):2105–2152.
- Griliches, Z. (1977). Estimating the returns to schooling: Some econometric problems. *Econometrica: Journal of the Econometric Society*, pages 1–22.
- Heckman, J. and Vytlacil, E. (1998). Instrumental variables methods for the correlated random coefficient model: Estimating the average rate of return to schooling when the return is correlated with schooling. *Journal of Human Resources*, pages 974–987.
- Hoderlein, S., Holzmann, H., and Meister, A. (2017). The triangular model with random coefficients. *Journal of Econometrics*, 201(1):144–169.
- Hoderlein, S., Klemelä, J., and Mammen, E. (2010). Analyzing the random coefficient model nonparametrically. *Econometric Theory*, 26(3):804–837.
- Hsieh, C.-T. and Klenow, P. J. (2009). Misallocation and manufacturing tfp in china and india. *The Quarterly journal of economics*, 124(4):1403–1448.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.
- Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer.
- Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Koenker, R. (2004). Quantile regression for longitudinal data. *Journal of Multivariate Analysis*, 91(1):74–89.
- Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, 15(4):143–156.

- Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.
- LaPointe Rudow, D., Hays, R., Baliga, P., Cohen, D. J., Cooper, M., Danovitch, G. M., Dew, M. A., Gordon, E. J., Mandelbrot, D. A., McGuire, S., et al. (2015). Consensus conference on best practices in live kidney donation: recommendations to optimize education, access, and care. *American Journal of Transplantation*, 15(4):914–922.
- Lattimore, T. and Szepesvari, C. (2018). Bandit algorithms.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.
- Levinsohn, J. and Petrin, A. (2003). Estimating production functions using inputs to control for unobservables. *The Review of Economic Studies*, 70(2):317–341.
- Lewbel, A. (2016). The identification zoo-meanings of identification in econometrics. *Forthcoming on Journal of Economic Literature*, 7.
- Marschak, J. and Andrews, W. H. (1944). Random simultaneous equations and the theory of production. *Econometrica, Journal of the Econometric Society*, pages 143–205.
- Montgomery, R. A., Gentry, S. E., Marks, W. H., Warren, D. S., Hiller, J., Houp, J., Zachary, A. A., Melancon, J. K., Maley, W. R., Rabb, H., et al. (2006). Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet*, 368(9533):419–421.
- Mundlak, Y. (1978). Models with variable coefficients: integration and extension. In *Annales de l'INSEE*, pages 483–509. JSTOR.
- Organ Procurement and Transplantation Network (2013). Current cpra calculation, updated 2013.
- Popoviciu, T. (1935). Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9:129–145.
- Rapaport, F. T. (1986). The case for a living emotionally related international kidney donor exchange registry. In *Transplantation proceedings*, volume 18, pages 5–9.
- Rees, M. A., Kopke, J. E., Pelletier, R. P., Segev, D. L., Rutter, M. E., Fabrega, A. J., Rogers, J., Pankewycz, O. G., Hiller, J., Roth, A. E., et al. (2009). A non-simultaneous, extended, altruistic-donor chain. *New England Journal of Medicine*, 360(11):1096–1101.
- Roth, A. E. (2015). *Who Gets What and Why: The New Economics of Matchmaking and Market Design*. Houghton Mifflin Harcourt.
- Roth, A. E., Sönmez, T., and Ünver, M. U. (2004). Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488.

- Roth, A. E., Sönmez, T., and Ünver, M. U. (2005). Pairwise kidney exchange. *Journal of Economic theory*, 125(2):151–188.
- Roth, A. E., Sönmez, T., and Ünver, M. U. (2007). Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851.
- Russo, D., Van Roy, B., Kazerouni, A., and Osband, I. (2017). A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*.
- Saidman, S. L., Roth, A. E., Sönmez, T., Ünver, M. U., and Delmonico, F. L. (2006). Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation*, 81(5):773–782.
- Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of mathematical economics*, 1(1):23–37.
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98.
- Slivkins, A. (2018). Introduction to multi-armed bandits. Available at <http://slivkins.com/work/MAB-book.pdf>. Retrieved on Jan 11, 2018.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Swamy, P. A. (1970). Efficient inference in a random coefficient regression model. *Econometrica: Journal of the Econometric Society*, pages 311–323.
- Syverson, C. (2004). Product substitutability and productivity dispersion. *Review of Economics and Statistics*, 86(2):534–550.
- Syverson, C. (2011). What determines productivity? *Journal of Economic literature*, 49(2):326–65.
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.
- Tonelli, M., Wiebe, N., Knoll, G., Bello, A., Browne, S., Jadhav, D., Klarenbach, S., and Gill, J. (2011). Systematic review: kidney transplantation compared with dialysis in clinically relevant outcomes. *American journal of transplantation*, 11(10):2093–2109.
- Ünver, M. U. (2010). Dynamic kidney exchange. *The Review of Economic Studies*, 77(1):372–414.

- Wooldridge, J. M. (2005). Fixed-effects and related estimators for correlated random-coefficient and treatment-effect panel data models. *Review of Economics and Statistics*, 87(2):385–390.
- Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press.
- Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721.
- Zenios, S., Woodle, E. S., and Ross, L. F. (2001). Primum non nocere: avoiding increased waiting times for individual racial and blood-type subsets of kidney wait list candidates in a living donor/cadaveric donor exchange program. *Transplantation*, 72(4):648–654.