A Bayesian Optimization Approach to Estimating Expected Match Time and Organ Quality in Kidney Exchange

Naveen Durvasula[†] and **Aravind Srinivasan**[‡] and **John P. Dickerson**[‡]
[†]Montgomery Blair High School [‡]University of Maryland
140.naveen.d@gmail.com, {srin,john}@cs.umd.edu

Abstract

Kidney exchanges allow patients with end-stage renal disease to find a lifesaving living donor by way of an organized market. However, not all patients are equally easy to match, nor are all donor organs of equal quality—some patients are matched within weeks, while others may wait for years with no match offers at all. Knowledge of expected waiting time and organ quality affects medical and insurance decisions. This work presents a principled method to estimate the expected quality of the kidney that a specific patient who enters an exchange will receive, as well as how long it will take to find that match. Estimation is performed via a novel Bayesian-optimization-based approach that learns a model of a computationally complex underlying Monte Carlo simulator. With a limited number of expensive simulation trajectories, the model produces practically-applicable results. Such fast and accurate sampling could provide medical professionals near-instantaneous access to valuable insight regarding a patient's expected outcome in a kidney exchange system.

1 Introduction

Renal disease affects millions of people worldwide, with a societal burden comparable to diabetes (Neuen et al. 2013). A patient with end-stage renal failure requires one of two treatments to stay alive: frequent and costly filtration & replacement of their blood (dialysis), or the reception of an organ transplant from a donor with one or more healthy kidneys. The latter option is often preferable due to increased quality of life and other health outcomes (Santos et al. 2015). Donor kidneys are obtained from one of three sources: the deceased donor waiting list, where cadaveric kidneys are harvested from deceased donors with still-healthy kidneys; ad-hoc arrangements between a compatible living donor and a patient; and, recently, *kidney exchanges* – an organized market where patients swap willing donors with other patients (Roth, Sönmez, and Ünver 2004; 2005a; 2005b). Kidney exchanges, while still quite new, result in increased numbers and quality of transplants (Sönmez, Unver, and Yenmez 2017); furthermore, their design is a success story for fielded AI research (Abraham, Blum, and Sandholm 2007; Ashlagi and Roth 2014; Anderson et al. 2015; Dickerson and Sandholm 2015; Hajaj et al. 2015; Toulis and Parkes 2015; Manlove and O'Malley 2015).

The act of getting a kidney transplant is time-sensitive, and affects healthcare and lifestyle decisions; furthermore, the expected quality of the kidney—if any—received by a patient affects the decision to accept or reject a particular match offer, and may be used to (de)prioritize patients in a matching mechanism (Bertsimas, Farias, and Trichakis 2013). Thus, decision-support systems that incorporate donor and patient features and quantify or predict the value of a current or future offered kidney are valuable to practitioners. The Kidney Donor Profile Index (KDPI) (Rao et al. 2009) and the Living Kidney Donor Profile Index (LKDPI) (Massie et al. 2016) are well-known and used to assess deceased- and living-donor kidneys, respectively. However, no method (nor system) currently exists to find the expected quality of a donated kidney in a kidney exchange.

This paper presents a Bayesian-optimization-based system that takes as input features of a patient and their paired donor, and returns an estimate of (i) the expected quality of a match, and (ii) expected waiting time for a matched kidney offer. The use of modern tools from machine learning and combinatorial optimization is required due to the NP-hard and APX-hard nature of even the most basic problems in kidney exchange (Abraham, Blum, and Sandholm 2007; Biró and Cechlárová 2007; Biró, Manlove, and Rizzi 2009; Luo et al. 2016; Jia et al. 2017). Our method uses a realistic but expensive black-box Monte Carlo simulator to produce estimates of match quality and time-to-match for a specific patient and donor; it samples new points in the space intelligently, balancing overall computational time with the accuracy of prediction for a new patient and donor. This prediction can be done in real or near-real time, a requirement for such a decision-support system. We give a proof-of-concept implementation on a reduced but realistic set of features in the kidney-exchange setting, and show that the method learns the necessary functions well.

2 Preliminaries

We now overview terminology for the allocation of kidneys (§2.1) and the standard model of kidney exchange (§2.2).

2.1 Deceased- & Living-Donor Kidney Allocation

Our motivation in this paper is, in part, due to the widespread usage of the Kidney Donor Profile Index (KDPI) to quantify the value of deceased-donor kidneys, and the increasing use of the new Living Kidney Donor Profile Index (LKDPI) to quantify the value of living-donor kidneys (Rao et al. 2009; Massie et al. 2016). Roughly speaking, both the KDPI and

the LKDPI are metrics used to compute the expected lifetime (quality) of a kidney that is donated from a donor D to a patient P. Both are based on multivariate Cox Regression models adapted from the traditional statistics literature (Cox 1992). The LKDPI metric was constructed such that LKDPI scores can be directly compared with KDPI scores, thus allowing direct comparison between living donor and deceased donor options. We expand this metric of quality to fielded kidney exchange. Unlike a standard ad-hoc living-donor donation, in a donation through a kidney exchange, the features of the end donor are unknown, and are generated through a stochastic matching process. We aim to compute the expected LKDPI of the kidney received through kidney-paired donation, and the expected matching time that it would take to receive this kidney, in order to allow for comparison between the living donor, deceased donor, and kidney-paired donation options. Because we build on the LKDPI metric here, we restate its calculation next:

$$\begin{aligned}\text{LKDPI}(D,P) = &-11.30\\ &+1.85*[(D_{\textbf{age}}-50)\ \textit{if}\ D_{\textbf{age}}>50]\\ &-0.381*D_{\textbf{eGFR}}\\ &+1.17*D_{\textbf{BMI}}\\ &\quad(+22.34\ \textit{if}\ D_{\textbf{is African-American}})\\ &\quad(+14.33\ \textit{if}\ D_{\textbf{has history of cigarette use}})\\ &+0.44*D_{\textbf{systolic blood pressure}}\\ &\quad(-21.68\ \textit{if}\ D\ \textbf{and}\ P\ \textbf{are both male})\\ &\quad(+27.30\ \textit{if}\ D\ \textbf{and}\ P\ \textbf{are ABO incompatible})\\ &\quad(-10.61\ \textit{if}\ D\ \textbf{and}\ P\ \textbf{are unrelated})\\ &+8.57*(\#\textbf{HLA-B mismatches})\\ &+8.26*(\#\textbf{HLA-DR mismatches})\\ &-50.87*\left[\min\left(\frac{D_{\textbf{weight}}}{P_{\textbf{weight}}},0.9\right)\right]\end{aligned} \tag{1}$$

Here, the estimated glomerular filtration rate (eGFR), body mass index (BMI), blood type (ABO) compatibility, and human leukocyte antigen (HLA) are all integral or real values determined by physical medical testing.[1]

2.2 The Formal Kidney-Exchange Model

The most-used model represents a kidney exchange as a directed graph $G = (V, E)$, called a *compatibility graph*. Here, each patient and their paired donor who enter the pool are represented as a *single* vertex. Then, a directed edge is drawn from vertex v_i to vertex v_j if the patient at vertex v_j wants the donor kidney of vertex v_i. Weights $w_e \in \mathbb{R}$ represent the utility of an individual kidney transplant represented by an edge e, and are also used to (de)prioritize specific classes of patient (Dickerson, Procaccia, and Sandholm 2014; UNOS 2015).

[1]While we direct the reader to the medical literature for a full explanation of all variables in this equation (Massie et al. 2016), we overview ABO compatibility here. At a high level, blood is partitioned into four types: O, A, B, and AB. Blood type O, known as the "universal donor" type, can be donated to patients of any other blood type. Blood types A and B can be donated to types A and B, respectively, along with type AB; blood type AB can be donated only to those of type AB. A donor whose blood type can be donated to a patient is said to be an *ABO compatible* donor.

Kidney exchanges rely on one of two types of structures to match patients: cycles and chains. First, a k-*cycle* c consists of exactly k patient-donor pairs (vertices), each connected by an edge in a cycle; here, each pair in c receives the kidney from the previous pair. Second, a k-*chain* begins with a non-directed donor, who enters the pool without a patient and gives their kidney to a patient with a paired donor, who gives to another patient with a paired donor, and so on k times.[2] Modern exchanges derive the majority of their utility from chains (Montgomery et al. 2006; Rees et al. 2009; Anderson et al. 2015; Ashlagi et al. 2017).

A *matching* M is a set of disjoint cycles and chains in a compatibility graph G; $M \in \mathcal{M}$, the set of all legal matchings. No donor can give more than one of her kidneys, necessitating the disjointness of cycles and chains—although recent work explores multi-donor donation (Ergin, Sönmez, and Ünver 2017; Farina, Dickerson, and Sandholm 2017). Given the set of all legal matchings $\mathcal{M}$, the *clearing problem* finds the matching M^* that maximizes utility function $u : \mathcal{M} \to \mathbb{R}$ (e.g., for maximum weighted matching, $u(M) = \sum_{c\in M}\sum_{e\in c} w_e$). Formally: $M^* \in \arg\max_{M\in\mathcal{M}} u(M)$. Ongoing research in the AI/Economics literature uses utility functions to enforce incentive properties via mechanism design (Ashlagi and Roth 2014; Li et al. 2014; Hajaj et al. 2015; Blum et al. 2017; Mattei, Saffidine, and Walsh 2017).

Finding a maximum weight (capped-length) cycle and chain packing is NP-hard (Abraham, Blum, and Sandholm 2007; Biró, Manlove, and Rizzi 2009), and is also hard to approximate (Biró and Cechlárová 2007; Luo et al. 2016; Jia et al. 2017). In practice, integer program (IP) formulations are used to clear large exchanges (Abraham, Blum, and Sandholm 2007; Dickerson, Procaccia, and Sandholm 2013; Glorie, van de Klundert, and Wagelmans 2014; Anderson et al. 2015; Dickerson et al. 2016). Formally, denote the set of all legal chains of length at most K and cycles of length at most L by $\mathcal{C}(L, K)$. Then, solve the IP: $\max \sum_{c\in\mathcal{C}(L,K)} w_c\, x_c$ subject to $\sum_{c:v\in c} x_c \le 1 \quad \forall v \in V$, where $x_c \in \{0, 1\}$ is a variable for every $c \in \mathcal{C}(L, K)$, and $w_c = \sum_{e\in c} w_e$. The final matching is the set of chains and cycles c such that $x_c = 1$. We use this IP as a sub-solver.

The kidney-exchange system is dynamic. In each iteration, new patients enter the pool, current patients may leave due to competition from other methods for receiving a kidney or death, and edges may appear or disappear based on the health characteristics of participants (e.g., pregnancy or sickness, leading to a change in compatibility with potential donors). We simulate this complex dynamic process in our work; additional details are given in Section 4. The matching process is highly stochastic. In fielded kidney exchanges, matches are made without detailed knowledge of compatibility between a donor and patient. More-thorough physical *crossmatch tests* are done after an algorithmic match, but before the actual transplantation event, to ensure that a matched donor can donate to a paired patient. Even one fail-

[2]In fielded kidney exchanges, cycles are limited in size to, typically, 3; all surgeries in a cycle must be executed simultaneously, so longer cycles are nearly impossible to plan. Chains, however, can be much longer (or effectively endless) in practice.

ure of an edge in a cycle invalidates the *entire* cycle; similarly, given the incremental execution of chains, all potential transplants located after the first edge failure in a chain are invalidated. This stochasticity, in addition to the dynamic process of patients and donors entering and leaving the system, introduces significant noise into the system.

3 Estimating Functions of Expensive Stochastic Processes

We now specify our model for quantifying the quality of an organ, as well as our method for predicting the expected quality of that organ given a black box exchange simulator. As defined, the model and method are applicable to any function of a stochastic simulatable process; however, given our application, we motivate the model in the context of kidney exchange. We employ kidney exchange as a running example to explain notation and concepts.

Consider a *stochastic process* $\mathbb{S}(\mathbf{I}) \rightarrow \mathbf{M}$ which produces output samples $\mathbf{M}$ given some input features $\mathbf{I}$. Let $\mathbf{I} \in \mathcal{I}, \mathbf{M} \in \mathcal{M}$, where $\mathcal{I}$ denotes the *input space* and $\mathcal{M}$ denotes the *output space*.

Example 1 *Let $\mathbb{S}$ denote the kidney-exchange process, briefly described in Section 2, and defined in depth in Section 4. We compute samples from $\mathbb{S}$ by constructing a realistic simulator. Let $\mathbf{I}$ denote the feature set of a patient-donor pair that enters the system. Let $\mathbf{M}$ be a set containing the feature set of the donor that gets matched to the patient of $\mathbf{I}$, as well as the total time for $\mathbf{I}$ to be matched.*

We say a function f has *additive structure* if, for some $k \in \mathbb{N}$, it can be decomposed into constituent functions f_k such that $f(\mathbf{x}) = \sum_k f_k(\mathbf{x})$, for all valid inputs $\mathbf{x}$. Note that by this definition, all functions are additive for $k = 1$.

Consider some stochastic additive function $f(\mathbf{I}, \mathbf{M}|\mathbf{I})$, with $\mathbf{M}$ and $\mathbf{I}$ as defined above. The notation $\mathbf{M}|\mathbf{I}$ is used to show that $\mathbf{M}$ is not a real input to the function—the function depends only on $\mathbf{I}$ as $\mathbf{M} = \mathbb{S}(\mathbf{I})$. However, to compute f, we require samples of $\mathbf{M}$. Let $F(\mathbf{I}) = \mathrm{E}[f(\mathbf{I}, \mathbf{M}|\mathbf{I})]$.

Example 2 *In our running example, assign $f(\mathbf{I}, \mathbf{M}|\mathbf{I}) := \mathrm{LKDPI}(\mathbf{M}|\mathbf{I}, \mathbf{I})$. That is, f simulates an exchange system and returns the* LKDPI *of the kidney that the patient of the patient-donor pair $\mathbf{I}$ receives through that matching process. Note that by the* LKDPI *formula, f is an additive function. Thus, $F(\mathbf{I})$ returns $\mathrm{E}[\mathrm{LKDPI}(\mathbf{M}|\mathbf{I}, \mathbf{I})]$. Let $t(\mathbf{I})$ be a stochastic function that returns the matching time for patient-donor pair $\mathbf{I}$, and $T(\mathbf{I}) = \mathrm{E}[t(\mathbf{I})]$. While, due to the complexity of the underlying exchange process, the t function does not decompose as easily into additive components, we treat it as an additive function with only one component. We aim to quickly—e.g., while waiting in a doctor's office—approximate, and provide patients, with accurate estimates of expected quality $F(\mathbf{I})$ and expected matching time $T(\mathbf{I})$, given membership in a patient-donor pair with features $\mathbf{I}$.*

The naïve method to estimate the expected value F of such a function f is to, for every $\mathbf{I}$, to simulate $\mathbb{S}(\mathbf{I})$ for a sufficient number of trajectories, and take a sample average as an estimate of $F(\mathbf{I})$. However, this becomes intractable quickly if $\mathbb{S}$ simulates a sufficiently noisy and/or time-consuming process. In our setting, output samples of $\mathbf{M}$ are very noisy, and even the basic matching policy described in Section 2 is intractable if a sufficient number of samples are taken to accurately compute $F(\mathbf{I})$ and $T(\mathbf{I})$. This estimation strategy becomes even less feasible if more-advanced dynamic matching policies that use reinforcement learning methods (Dickerson and Sandholm 2015) are used, which increase simulation (and thus sample) time.

We place no constraints on the f_k above: thus, trying to learn *discontinuous* F directly is hard. We note that F also has an additive structure. We exploit this additive structure to effectively learn F: by the linearity of expectation, $F(\mathbf{I}) = \mathrm{E}[f(\mathbf{I}, \mathbf{M}|\mathbf{I})] = \mathrm{E}\left[\sum_k f_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k)\right] = \sum_k \mathrm{E}[f_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k)] = \sum_k F_k(\mathbf{I}_k)$. Here, $\mathbf{M}_k \subseteq \mathbf{M}, \mathbf{I}_k \subseteq \mathbf{I}$, and both $\mathbf{I}_k$ and $\mathbf{M}_k$ are of the smallest size possible. Note that while elements of $\mathbf{M}_k$ can be reduced to only those that are required for computing f_k, the elements of $\mathbf{I}_k$ must be sufficient to compute $\mathbf{M}_k$ and f_k. In other words, if $\mathbf{I}_k \neq \mathbf{I}$, it must be a property of $\mathbb{S}$ that the elements of $\mathbf{I} \setminus \mathbf{I}_k$ do not alter the joint distribution $p(\mathbf{M}_k)$ over the elements of $\mathbf{M}_k$. Not all F_k are equally difficult to estimate. Consider the case where no elements of $\mathbf{I}$ change $p(\mathbf{M}_k)$. If this occurs, we can estimate F_k by first estimating $p(\mathbf{M}_k)$ by sampling from $\mathbb{S}$ and integrating

$$F_k(\mathbf{I}_k) = \int_{\mathbf{M}_k} \cdots \int f_k\left(\mathbf{I}_k, \bigcup_i \mathbf{M}_k^i\right) p\left(\bigcup_i \mathbf{M}_k^i\right) \prod_{i=1}^{l} d\mathbf{M}_k^i \tag{2}$$

where $\mathbf{M}_k = \bigcup_i \mathbf{M}_k^i$. However, F_k need not satisfy this strict condition, as is in our application and many others. Next, we present a Bayesian-optimization-based technique to learn F_k which uses $\mathbb{S}$ to generate noisy samples, learns a model of F_k, and then provides (after learning the model) a fast approximate-sampling capability to the end-user.

3.1 A Bayesian-Optimization-Based Approach

Bayesian optimization (BO) utilizes Gaussian processes (GP) to maximize an unknown function—in our case, the expected output of a realistic, but noisy and expensive to run, simulator of a real-world process. In BO, an *acquisition function* is used to select the next point at which to sample that would, typically, lead to an accurate estimate of the maximum of that function. Examples include Expected Improvement (EI), Lower Confidence Bound (LCB), and Maximum Probability of Improvement (MPI); Brochu et al. (Brochu, Cora, and De Freitas 2010) give an in-depth overview of techniques. A kernel (covariance) function is used to interpolate between known values of the function, and determine the confidence at each point. While BO offers a method to maximize a function where getting output is time consuming, we learn F_k using this method by taking as output the GP of the BO. We use the Bayesian optimization framework to choose informative values of $\mathbf{I}_k$ which will increase our understanding of F_k. More formally, let

$$f_k \sim \mathcal{GP}(\mu(\mathbf{I}_k), \mathrm{k}(\mathbf{I}_k, \mathbf{I}_k')), \tag{3}$$

where $\mathrm{k}(\mathbf{I}_k, \mathbf{I}_k')$ is a kernel (covariance) function. To learn a function using BO, we let the acquisition function $A(\mathbf{I}_k)$

return the variance at the point. By equation (3), we have

$$p(f_k(\mathbf{I}_k)) = \mathcal{N}(\mu_{\mathbf{I}_k}, \sigma^2_{\mathbf{I}_k}) \tag{4}$$

We let $A(\mathbf{I}_k) = \sigma^2_{\mathbf{I}_k}$, and optimize A using the limited-memory BFGS optimization algorithm (Andrew and Gao 2007). It then follows that $F_k(\mathbf{I}_k) = \mu_{\mathbf{I}_k}$.

Accounting for Categorical Variables. The input $\mathbf{I}_k$ may contain some categorical variables (e.g., ABO blood types take one of four values). Because of this, we cannot directly apply a GP over $\mathbf{I}_k$, as we cannot fit a kernel over these categorical variables. Instead, for each permutation of categorical variables, we perform the following scheme over $\mathrm{C}(\mathbf{I}_k)$, the subset of $\mathbf{I}_k$ containing only non-categorical variables. We let $\mathrm{D}(\mathbf{I}_k)$ be the subset containing only the categorical variables. We can compute F_k as

$$F_k(\mathbf{I}_k) = \mu^{\mathrm{D}(\mathbf{I}_k)}_{\mathrm{C}(\mathbf{I}_k)}, \tag{5}$$

where the notation $\mu^{\mathrm{D}(\mathbf{I}_k)}_{\mathrm{C}(\mathbf{I}_k)}$ denotes the mean at the point $\mathrm{C}(\mathbf{I}_k)$ on the GP conditioned on the categorical variables being set to $\mathrm{D}(\mathbf{I}_k)$.

Piecewise Functions. A special class of functions f_k for which a GP is not well-suited to model are piecewise functions. Let f_k be a piecewise function with c cases.

$$f_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k) = \begin{cases} f^1_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k), & \textit{if } C^1_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k) \\ f^2_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k), & \textit{if } C^2_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k) \\ \vdots & \vdots \\ f^c_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k), & \textit{otherwise} \end{cases}$$

where each function $C^i_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k)$ is a mutually-exclusive Boolean-valued function. We use the above scheme to learn each of the $F^i_k(\mathbf{I}_k) = \mathrm{E}[f^i_k(I_k, M_k)]$ as well as $\Pr[C^i_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k)]$ for $1 \leq i \leq c$. We compute F_k as

$$F_k(\mathbf{I}_k) = \sum_{i=1}^{c} \Pr[C^i_k(\mathbf{I}_k, \mathbf{M}_k|\mathbf{I}_k)] F^i_k(\mathbf{I}_k) \tag{6}$$

by linearity of expectations.

Example 3 *The* LKDPI *function (1) has several terms defined by piecewise function. One example is the term containing a donor's BMI. Let the condition* ***AA*** *denote that the donor is African-American. Let the condition* ***HCU*** *denote that the donor has a history of cigarette use. We can write this term as*

$$f_3(\mathbf{I}_3, \mathbf{M}_3) = \begin{cases} 1.17 * (D_{\textbf{\textit{BMI}}} + 36.67), & \textit{if } \textbf{\textit{AA}} \wedge \textbf{\textit{HCU}} \\ 1.17 * (D_{\textbf{\textit{BMI}}} + 22.34), & \textit{if } \textbf{\textit{AA}} \\ 1.17 * (D_{\textbf{\textit{BMI}}} + 14.33), & \textit{if } \textbf{\textit{HCU}} \\ 1.17 * D_{\textbf{\textit{BMI}}}, & \textit{otherwise} \end{cases}$$

Defining the Kernel. For this application, we use the radial basis function (RBF) kernel for all GPs, given by

$$\mathrm{k}(\mathbf{I}_k, \mathbf{I}'_k) = \exp\left(-\frac{\|\mathbf{I}_k - \mathbf{I}'_k\|^2}{2\ell^2}\right). \tag{7}$$

The kernel is only accurate if the ℓ length-scale hyperparameter is properly tuned. We use Hamiltonian Monte Carlo (HMC) to approximate the probability distribution over ℓ given the data that we have seen so far (Duane et al. 1987). HMC is a physics-inspired Markov Chain Monte Carlo (MCMC) technique. As the underlying distribution over ℓ is inaccessible, we construct a Markov chain which has an equilibrium distribution equal to the target distribution over ℓ. Sampling from this Markov chain allows us to efficiently explore the *typical set* of the underlying distribution. What makes HMC different from other MCMC methods such as Metropolis-Hastings is its ability to smoothly explore the typical set without failing due to random-walk behavior (Betancourt 2017). HMC accomplishes this by defining a vector field over the underlying distribution. Just taking the gradient of the distribution gives a vector field that points towards the mode of the distribution. However, we want to make this vector field guide the Markov chain through the typical set. This is analogous to a physical system where we have a gravitational center, and we are trying to determine the *momentum* required to carry an object into orbit. We redefine our terminology in terms of this analogy. We let our *position* $\mathbf{q}$ be given by a coordinate of the density function. In our implementation of HMC, $\mathbf{q} = \ell$. The momentum parameter $\mathbf{p}$ is defined as an auxiliary parameter over our density function $p(\mathbf{q})$, and has the same dimension as $\mathbf{q}$. We call this augmented space over $\mathbf{q}$ and $\mathbf{p}$ the *phase space*.

We choose a conditional distribution $p(\mathbf{p} \mid \mathbf{q})$ because we can marginalize out the momentum, as

$$p(\mathbf{q}, \mathbf{p}) = p(\mathbf{p} \mid \mathbf{q}) p(\mathbf{q})$$

We can write this conditional distribution in terms of an invariant *Hamiltonian* function $H(\mathbf{q}, \mathbf{p})$ as

$$p(\mathbf{q}, \mathbf{p}) = e^{-H(\mathbf{q}, \mathbf{p})} \tag{8}$$

We refer to the value of the Hamiltonian at a given point within the phase space as the *energy* at that point. We get

$$\begin{aligned} H(\mathbf{q}, \mathbf{p}) &= -\log(p(\mathbf{q}, \mathbf{p})) \\ &= -\log(p(\mathbf{p} \mid \mathbf{q})) - \log(p(\mathbf{q})) \\ &\equiv K(\mathbf{p}, \mathbf{q}) + V(\mathbf{q}) \end{aligned} \tag{9}$$

We refer to K as the *kinetic energy*, and V as the *potential energy*. We can generate a vector field oriented with the typical set using *Hamilton's equations*, given by

$$\begin{aligned} \frac{d\mathbf{p}}{dt} &= \frac{\partial H}{\partial \mathbf{p}} = \frac{\partial K}{\partial \mathbf{p}} \\ \frac{d\mathbf{q}}{dt} &= -\frac{\partial H}{\partial \mathbf{q}} = \frac{\partial K}{\partial \mathbf{q}} - \frac{\partial V}{\partial \mathbf{q}} \end{aligned} \tag{10}$$

We let

$$p(\mathbf{p} \mid \mathbf{q}) = \mathcal{N}(\mathbf{p} \mid 0, M) \tag{11}$$

and thus define a *Euclidean-Gaussian* kinetic energy

$$K(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\mathbf{p}^\intercal M^{-1}\mathbf{p} + \log|M| + \text{const.}, \tag{12}$$

where M is defined by rotating and scaling a matrix of Euclidean inner products. For more information, please see Betancourt (Betancourt 2017). HMC can be used to improve

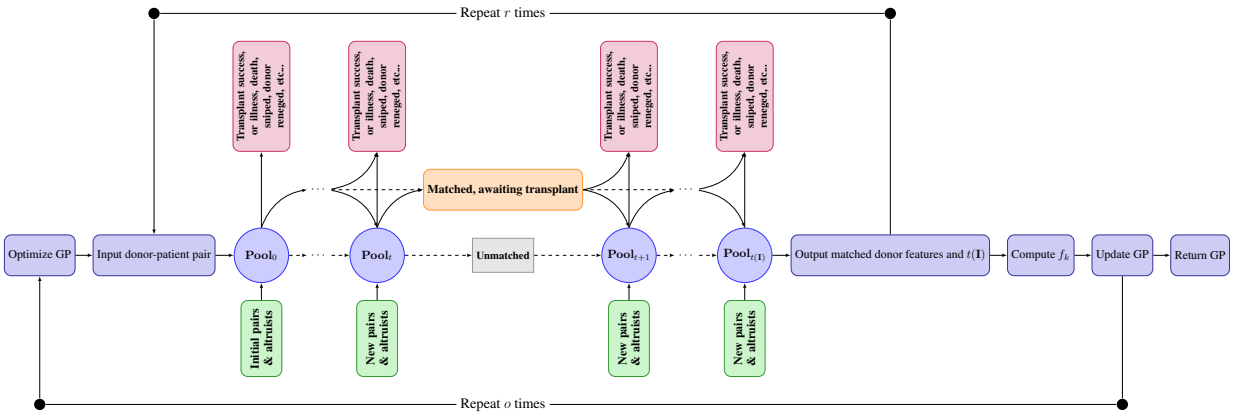

Figure 1: Estimating F_k or $T(\mathbf{I})$ using Bayesian optimization

the performance of GP-based applications in general. Standard length-scale tuning methods involve optimizing $p(\ell)$ to find the maximum likelihood value. However, as in (Betancourt 2017), the maximum-likelihood value can lie far from elements of the typical set. (Appendix A shows experimentally the negative impact of this on estimation quality.) Thus, we modify our acquisition function A to compute the expected value $\mathrm{E}[\sigma^2_{\mathbf{I}_k}]$ over the kernel's ℓ distribution:

$$A(\mathbf{I}_k) = \int_0^\infty (\sigma^2_{\mathbf{I}_k}|\ell)p(\ell)d\ell \tag{13}$$

4 Estimating Match Time and Quality: Evidence from a Realistic Simulation

In this section, we return to the initial motivation for this paper—creating a decision support system that quickly and accurately estimates to a patient and paired donor the expected waiting time and offered organ quality. Coupled with the well-known and widely-used KDPI and LKDPI metrics currently fielded in the deceased-donor and non-exchange-based living-donor kidney programs, respectively, a principled translation of those techniques to the more complicated kidney-exchange setting has the potential for great impact. This section presents an exploratory validation of our approach in a reduced, but still realistic, model of kidney exchange. We begin by describing that model, including our feature space and simulator (§4.2), presenting quantitative results performed on commodity hardware (§4.3), and concluding with a general discussion (§4.4).

4.1 Experimental Setup

We instantiate the model under a simplified kidney-exchange process.[3] In our model, the only factors that affect $p(\mathbf{M})$ in this matching policy are a patient's blood type, their potential donor's blood type, their paired but incompatible donor's blood type, and the respective CPRA (described next) of all involved parties (patient, paired donor, potential donor). Here, the Calculated Panel Reactive Antibodies (CPRA) is a continuous-valued score in $[0, 1]$, roughly representing the fraction of donors, drawn from a general population, that would *not* be a match for a particular patient (e.g., a score of 1 signals extreme difficulty in matching). This simplified feature set implies that the only one of these factors which appear in the LKDPI formula is patient blood type. Furthermore, it only appears in the form of a comparison between the matched donor's blood type and the patient's blood type. $\mathcal{C}(L, K)$ as mentioned in Section 2 only contains cycles and chains where the matched donor has the same blood type as the patient. Thus, all F_k can be computed via (2). We evaluate the BO approach by estimating the expected matching time $T(\mathbf{I})$. Let $\mathcal{I}$ be the space over the features listed above. To reduce noise in the output of $\mathbb{S}$, we compute r samples of $\mathbb{S}(\mathbf{I})$ in each optimization iteration. We let matches take place on a weekly basis, and cap the total match-time at 250 weeks; each time period invokes a state-of-the-art IP-based codebase (Dickerson and Sandholm 2015) that solves the NP-hard optimization problem from Section 2.2.[4] We use this sub-solver to clear the kidney exchange pool at every time period, and simulate the evolution of the pool—vertices arriving and departing, edges existing and failing—as in the realistic-kidney-exchange simulation literature (Saidman et al. 2006; Anderson et al. 2015; Dickerson and Sandholm 2015). Fig. 1 describes the instantiation of the BO approach for LKDPI computation.

4.2 Bayesian Optimization for Match Time Computation

We use GPyOpt (GPyOpt 2016), an open-source Bayesian-optimization platform, for implementing our method. We modified GPyOpt to change the acquisition to return the integrated expected variance at the point as described in Section 3. Then, for all 16 blood-type-pair combinations for donor-patient pairs who enter the exchange, we performed Bayesian optimization over patient CPRA for $o = 50$ iter-

[3] For an in-depth discussion of all features that are currently included in a large, real-world kidney exchange due to the United Network for Organ Sharing (UNOS) in the USA, we direct the reader to a recent white-paper from that group (UNOS 2015).

[4] Code for that base simulator can be found here: `https://github.com/JohnDickerson/KidneyExchange`; after the double-blind review period concludes, we will make public our fork of their repository that includes all code to replicate experiments in this paper, including data generation and output parsing.

ations. For each blood-type-pair combination, $T(\mathbf{I})$ is estimated based on $r = 48$ trajectories in a realistic simulator.

We now experimentally validate the proposed method using a realistic kidney-exchange simulator and a reduced feature set, as a proof of concept. There exists no comparable baseline that can be used to validate our method. Instead, we show that our model is able to make clinically promising estimates of the expected waiting time even with noisy (but inexpensive) input. We further justify our model by comparing the performance of our model to that of several passive learning models in the Scikit-learn framework (Pedregosa et al. 2011). In practice, one would include 25-30 features before making a policy recommendation; however, as §4.3 will show, even using a reduced feature set validates the method.

After constructing the GPs through BO, we test them by comparing the match time returned by the GP and the match time returned by the realistic simulator after s trajectories. To test the 16 generated GPs, the domain of CPRA $[0, 1]$ is partitioned uniformly into 4 zones $\{[0, .25), [.25, .5), [.5, .75), [.75, 1)\}$. In each zone, 5 random trials are done, with $s = 128$ trajectories each. We test the system with $s > r$ in order to measure the ability of the model to make good estimations of the expected value even with more noisy input.

We compare our method to common regression models using the Scikit-learn framework. Each of these models was given as input 50 CPRA values selected uniformly at random on the domain $[0, 1]$ for each of the 16 blood type combinations. We perform 20 iterations of randomized hyperparameter optimization (see Appendix B for detailed information), endowing all relevant hyperparameters with appropriate exponential (or geometric, for discrete variables) distributions. We compare our method against the following models: random forest regression, isotonic regression, support vector regression (SVR) with RBF kernel, SVR with polynomial kernel, and multi-layer perceptron. Note that none of these models are capable of learning the underlying noise model of the function. In our application, *the noise model is of clinical importance* as it allows stakeholders (e.g., doctors, patients, insurance) to know the margin of error on the expected-waiting-time estimate.

4.3 Experimental Results

Figure 2 shows the estimated expected mean residual of our estimated function when compared to our simulated test of the value of that function in weeks, calculated over all blood type combinations.[5] First, the learned function fits the true (simulated) function quite well, for CPRA values far from 1—within two or three weeks of realized match time. As the CPRA value approaches 1, the residuals increase; this is to be expected, because higher CPRA values result in (much) more uncertainty in the ability to find any match at all. Indeed, a CPRA of 1 indicates zero probability of finding a match; in a case like this, waiting time becomes infinite, and the function to be estimated becomes ill-defined. To prevent the simulation from taking an indefinite amount of time, we cap the total match time at 250 weeks.

[5] Appendix C, Figure 10 presents that data in tabular format.

Figure 2 demonstrates that for the first 3 CPRA zones, comparative performance among all models tested is roughly equal. Due to the nature of the kidney exchange process, it happens to be that the CPRA zone in which the function becomes the most stochastic (zone 4) is also the subset of the domain in which function changes the most. Thus, it is in this CPRA zone in which we see varied performance. With the exception of isotonic regression, our method outperforms the passively learned models. However, we note that isotonic regression assumes that the function is monotonic and for the general F_k, this is not true. Interestingly, our method shows very similar performance to the isotonic regressor even without an assumption of monotonicity. Our method also learns an underlying noise model, which is of clinical importance.

Figure 3 shows the mean residual in weeks over CPRA for the donor-patient blood type combination O-AB. The O-AB pair is, in some sense, the easiest to match; its donor has the "universal" blood type O, and its patient has no blood type constraints. Indeed, we see that the model estimates the waiting time and quality of this "easy" donor-patient pair quite well. Residuals are small enough that they can be attributed to the stochasticity in the trials used to test the system.[6]

Figure 4 shows the mean residual in weeks over CPRA for the blood type combination AB-O—a "hard" donor-patient pair. Figure 5 shows the distribution on the function mapping patient CPRA for that blood type combination in weeks, and the point at which the acquisition is optimized (the solid vertical red line). We focus on this particular case because the AB-O pair is a complement to the O-AB pair; while the latter is easy to match, the former has the most constrained donor (type AB donors can only give to type AB patients), and the most constrained patient (type O patients can only receive from type O donors). This increased difficulty in matching led to increased stochasticity in the result, thus giving a higher mean residual of between 6 and 12 weeks.

We performed this experiment for all 16 donor-patient blood types. Figure 2 gives a high-level quantitative overview of those results; qualitatively, results followed this visualization. In general, higher CPRA leads to greater uncertainty. Similarly, harder-to-match (in terms of ABO blood type) patients as well as donors also lead to greater uncertainty in matching time. However, throughout, our method was able to learn a close approximation—typically within a few weeks of the actual matching time—to the underlying function of interest, all using commodity hardware.

4.4 Discussion

Overall, even with a limited amount of trajectories, the GP provides accurate predictions of the underlying noisy function. As a patient's CPRA increases, the mean residual increases, as the match time output increases in stochasticity with respect to an increase in CPRA. With harder-to-match blood types (e.g., the AB-type donor, O-patient donor patient pair), the stochasticity in the result was larger as

[6] Appendix C, Figure 11 shows the distribution on the estimated function mapping patient CPRA for the blood type combination O-AB. We see that uncertainty is quite low, even after only a small number of (expensive) black box function queries.

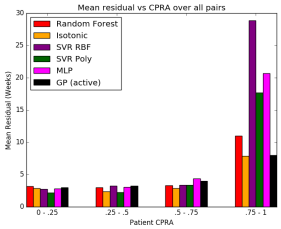

Figure 2: Mean absolute residual (weeks), all blood type pairs, vs. $s = 128$ estimates of expected waiting time.

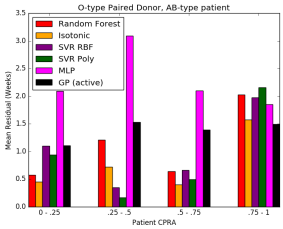

Figure 3: Mean absolute residual (weeks), O-AB donor-patient pair, vs. $s = 128$ estimates of expected waiting time.

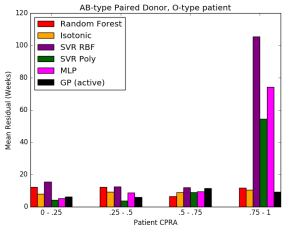

Figure 4: Mean absolute residual (weeks), AB-O donor-patient pair, vs. $s = 128$ estimates of expected waiting time.

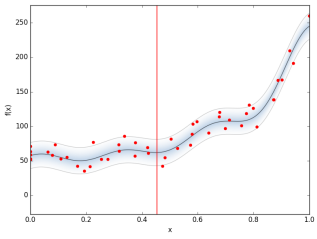

Figure 5: Our learned GP for AB-type paired donor and O-type patient (in weeks), varying patient CPRA (x-axis).

well, resulting in a higher mean residual. However, as seen in Fig. 4, the GP is capable of handling this, by estimating the probability distribution on that value (using a Gaussian prior). Indeed, Fig.s 2 and 10 demonstrate that the GP with actively-sampled input well outperforms the passively-learned models in the most stochastic CPRA zone with the exception of isotonic regression. However, isotonic regression applies only to monotonic functions. Thus, to a certain extent, the GP is capable of handling stochastic outputs. We note that both of these edge cases—patients with extremely high CPRA, said to be "highly sensitized," and donor-patient pairs whose blood types are hard to match (like the AB-O donor-patient pairs discussed above), and are said to be "under-demanded,"—have previously received, and continue to receive, special attention in both the economic and medical policy literature. For example, the economic literature suggests that AB-O pairs may *never* be matched in a fully efficient matching (Ashlagi and Roth 2014; Toulis and Parkes 2015), while similarly negative results exist for the (lack of) opportunity to match highly-sensitized patients under fully efficient matching policies (Dickerson, Procaccia, and Sandholm 2014; McElfresh and Dickerson 2018). Our IP-based approach maximizes (short-term) efficiency, so it is not surprising that the highest variance results for our system come from the hardest-to-match patients.

While our proof-of-concept demonstrates the promise of this system in a toy environment, before making a policy recommendation or deploying a support tool in practice, we note the following: (i) The number of trajectories r should be (much) greater, for greatly reduced stochasticity, and thus far smaller mean residuals from the GP to the realistic simulator. (ii) The number of features considered should be much higher, and informed by experts in the field. Yet, given these proof-of-concept experimental results, we feel confident that—given all F_k—a decision-support system can be deployed for use by practitioners, in order to give patients and their willing donors this information on demand.

5 Conclusions & Future Research

This paper presents a principled method to estimate the expected quality of the kidney that a specific patient who enters an exchange will receive, as well as how long it will take to find that match. Knowledge of expected waiting time and organ quality affects medical and insurance decisions. Estimation was performed via a novel Bayesian-optimization-based approach that learns a model of a computationally-complex Monte Carlo simulator, which in turn represents a potentially discontinuous function. The method presented generalizes to any function of a stochastic (and expensive) simulatable process. With a limited number of expensive simulation trajectories, the model produced reliably accurate results in our proof-of-concept setting, supporting further investigation. With access to fast and accurate sampling, medical professionals could have near-instantaneous access to valuable insights regarding a patient's expected outcome in a kidney exchange system. One research direction is to determine, via a combination of feature selection methods and expert opinion, the set of features necessary to completely characterize the expected waiting time and kidney quality function our method aims to learn. Our model and proof-of-concept experiments support more-intense computational experiments. Similarly, one might use the application of LKDPI presented here to design more socially-beneficial mechanisms in the face of strategic agents, perhaps incentivizing agents to perform in a specific way to increase social welfare; this area of research, particularly in the context of kidney exchange, is still open (Ashlagi and Roth 2014; Toulis and Parkes 2015; Hajaj et al. 2015; Blum et al. 2017).

References

Abraham, D.; Blum, A.; and Sandholm, T. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 295–304.

Anderson, R.; Ashlagi, I.; Gamarnik, D.; and Roth, A. E. 2015. Finding long chains in kidney exchange using the traveling salesman problem. *Proc. National Academy of Sciences*.

Andrew, G., and Gao, J. 2007. Scalable training of l_1-regularized log-linear models. In *International Conference on Machine Learning (ICML)*.

Ashlagi, I., and Roth, A. E. 2014. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics* 9(3):817–863.

Ashlagi, I.; Gamarnik, D.; Rees, M.; and Roth, A. E. 2017. The need for (long) chains in kidney exchange. Revision requested at Operations Research. Initial version appeared at ACM Conference on Electronic Commerce (EC-12).

Bertsimas, D.; Farias, V. F.; and Trichakis, N. 2013. Fairness,

efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research* 61(1):73–87.

Betancourt, M. 2017. A conceptual introduction to Hamiltonian Monte Carlo. *CoRR* abs/1701.02434.

Biró, P., and Cechlárová, K. 2007. Inapproximability of the kidney exchange problem. *Information Processing Letters* 101(5).

Biró, P.; Manlove, D. F.; and Rizzi, R. 2009. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications* 1(04):499–517.

Blum, A.; Caragiannis, I.; Haghtalab, N.; Procaccia, A. D.; Procaccia, E. B.; and Vaish, R. 2017. Opting into optimal matchings. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*.

Brochu, E.; Cora, V. M.; and De Freitas, N. 2010. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR* abs/1012.2599.

Cox, D. R. 1992. Regression models and life-tables. In *Breakthroughs in Statistics*. Springer. 527–541.

Dickerson, J. P., and Sandholm, T. 2015. FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI Conference on Artificial Intelligence (AAAI)*, 622–628.

Dickerson, J. P.; Manlove, D.; Plaut, B.; Sandholm, T.; and Trimble, J. 2016. Position-indexed formulations for kidney exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2013. Failure-aware kidney exchange. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 323–340.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2014. Price of fairness in kidney exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 1013–1020.

Duane, S.; Kennedy, A. D.; Pendleton, B. J.; and Roweth, D. 1987. Hybrid monte carlo. *Physics Letters B* 195(2):216–222.

Ergin, H.; Sönmez, T.; and Ünver, M. U. 2017. Multi-donor organ exchange. Working paper.

Farina, G.; Dickerson, J. P.; and Sandholm, T. 2017. Operation frames and clubs in kidney exchange. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Glorie, K. M.; van de Klundert, J. J.; and Wagelmans, A. P. M. 2014. Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing & Service Operations Management (MSOM)* 16(4).

GPyOpt. 2016. GPyOpt: A Bayesian optimization framework in Python.

Hajaj, C.; Dickerson, J. P.; Hassidim, A.; Sandholm, T.; and Sarne, D. 2015. Strategy-proof and efficient kidney exchange using a credit mechanism. In *AAAI Conference on Artificial Intelligence (AAAI)*, 921–928.

Jia, Z.; Tang, P.; Wang, R.; and Zhang, H. 2017. Efficient near-optimal algorithms for barter exchange. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 362–370.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Li, J.; Liu, Y.; Huang, L.; and Tang, P. 2014. Egalitarian pairwise kidney exchange: Fast algorithms via linear programming and parametric flow. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 445–452.

Luo, S.; Tang, P.; Wu, C.; and Zeng, J. 2016. Approximation of barter exchanges with cycle length constraints. *CoRR* abs/1605.08863.

Manlove, D., and O'Malley, G. 2015. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics* 19(1).

Massie, A. B.; Leanza, J.; Fahmy, L.; Chow, E.; Desai, N. M.; Luo, X.; King, E.; Bowring, M.; and Segev, D. 2016. A risk index for living donor kidney transplantation. *American Journal of Transplantation* 16(7):2077–2084.

Mattei, N.; Saffidine, A.; and Walsh, T. 2017. Mechanisms for online organ matching. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

McElfresh, D., and Dickerson, J. P. 2018. Balancing lexicographic fairness and a utilitarian objective with application to kidney exchange. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Montgomery, R.; Gentry, S.; Marks, W. H.; Warren, D. S.; Hiller, J.; Houp, J.; Zachary, A. A.; Melancon, J. K.; Maley, W. R.; Rabb, H.; Simpkins, C.; and Segev, D. L. 2006. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet* 368(9533):419–421.

Neuen, B. L.; Taylor, G. E.; Demaio, A. R.; and Perkovic, V. 2013. Global kidney disease. *The Lancet* 382(9900):1243.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Rao, P. S.; Schaubel, D. E.; Guidinger, M. K.; Andreoni, K. A.; Wolfe, R. A.; Merion, R. M.; Port, F. K.; and Sung, R. S. 2009. A comprehensive risk quantification score for deceased donor kidneys: the kidney donor risk index. *Transplantation* 88(2):231–236.

Rees, M.; Kopke, J.; Pelletier, R.; Segev, D.; Rutter, M.; Fabrega, A.; Rogers, J.; Pankewycz, O.; Hiller, J.; Roth, A.; Sandholm, T.; Ünver, U.; and Montgomery, R. 2009. A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine* 360(11):1096–1101.

Roth, A.; Sönmez, T.; and Ünver, U. 2004. Kidney exchange. *Quarterly Journal of Economics* 119(2):457–488.

Roth, A.; Sönmez, T.; and Ünver, U. 2005a. A kidney exchange clearinghouse in New England. *American Economic Review* 95(2):376–380.

Roth, A.; Sönmez, T.; and Ünver, U. 2005b. Pairwise kidney exchange. *Journal of Economic Theory* 125(2):151–188.

Saidman, S. L.; Roth, A.; Sönmez, T.; Ünver, U.; and Delmonico, F. 2006. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation* 81(5):773–782.

Santos, A. H.; Casey, M. J.; Wen, X.; Zendejas, I.; Rehman, S.; Womer, K. L.; and Andreoni, K. A. 2015. Survival with dialysis versus kidney transplantation in adult hemolytic uremic syndrome patients: A fifteen-year study of the waiting list. *Transplantation* 99(12):2608–2616.

Sönmez, T.; Unver, U.; and Yenmez, M. B. 2017. Incentivized kidney exchange. Tech. report, Boston College Dept. of Economics.

Toulis, P., and Parkes, D. C. 2015. Design and analysis of multi-hospital kidney exchange mechanisms using random graphs. *Games and Economic Behavior* 91(0):360–382.

UNOS. 2015. Revising kidney paired donation pilot program priority points. OPTN/UNOS Public Comment Proposal.

Appendix for Paper ID: 6191

A Optimizing $p(\ell)$ versus HMC Sampling: Impact on System Performance

It is common in GP regression to set ℓ to be the value of maximum likelihood. However, the expected length-scale can deviate far from the mode. We show some resulting GPs when we ran our experiment as described in Section 4 without using HMC and instead tuning ℓ using the value of maximum likelihood. Figures 6 and 7 show the poor prediction results that occur when the maximum likelihood value of ℓ falls far above the expectation.

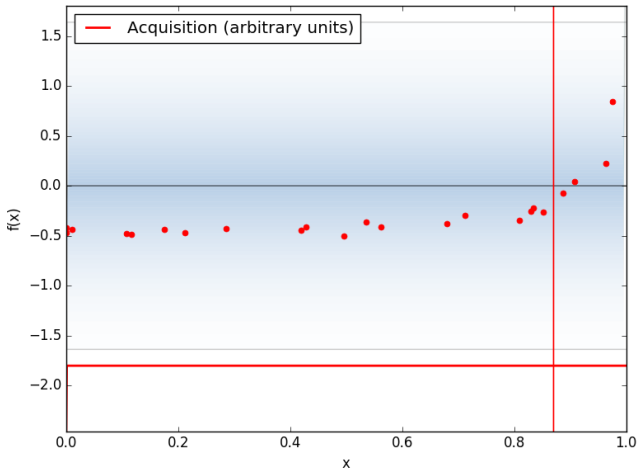

Figure 6: Setting ℓ too high, patient blood type B, donor blood type A.

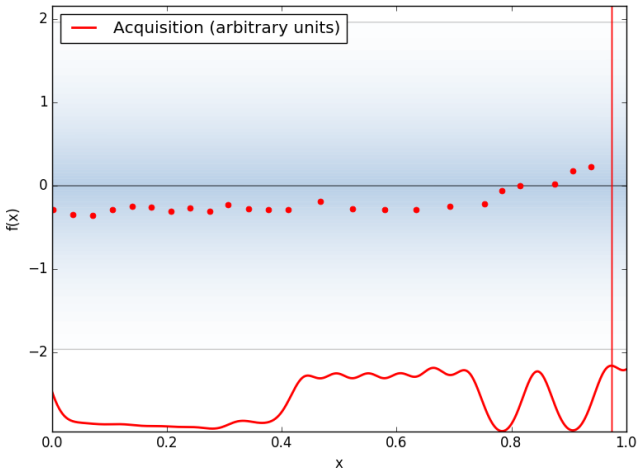

Figure 7: Setting ℓ too high, patient blood type AB, donor blood type A.

Likewise, Figures 8 and 9 show the negative effect on performance when the maximum likelihood value of ℓ falls far below the expectation.

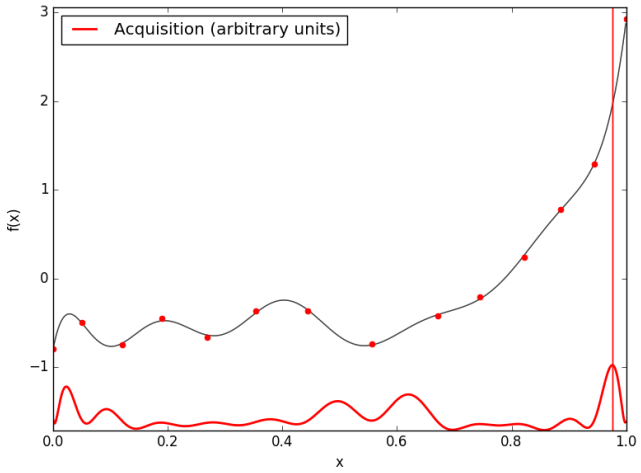

Figure 8: Setting ℓ too low, patient blood type B, donor blood type AB

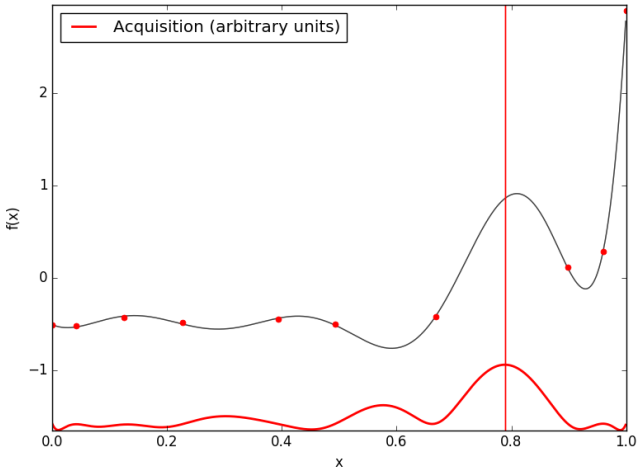

Figure 9: Setting ℓ too low, patient blood type AB, donor blood type B.

HMC allows us to efficiently compute the expectation, thus eliminating this issue.

B Hyperparameter Optimization

We describe in detail the hyperparameter optimization scheme that we used for the models that we used to compare our method against. We performed 20 search iterations of randomized hyperparameter optimization, where for relevant continuous (discrete) parameters, we sampled from an exponential (geometric) distribution with expected value approximately equal to the default value given by the Scikit-learn pipeline. We state these values below. Isotonic regression is not mentioned as there are no hyperparameters to optimize.

B.1 Random forest regression

We performed random forest regression geometrically distributing the number of regressors with expected value 20

B.2 Support vector regression

For both SVR regressors (RBF and Poly), we let C and γ, the error penalty parameter and kernel coefficient respectively, be exponentially distributed with expected value 1. Likewise, we let ϵ, the penalty tolerance constant, be exponentially distributed with expected value 0.1.

B.3 Multi-layer perceptron

We used the Adam optimizer (Kingma and Ba 2014) and hyperbolic tangent activation function. The MLP was trained for 1000 iterations. We set the learning rate to be constant, exponentially distributed with expected value 0.005. We let α, the L2 penalty regularization term, be exponentially distributed with expected value 0.0001. We let the total number of neurons be geometrically distributed with expected value 100.

C Additional Experimental Results

In this section, we present additional experimental results that accompany those presented in Section 4 of the main paper. In the main paper, Figure 2 showed the estimated expected mean residual of our estimated function when compared to our simulated test of the value of that function in weeks, calculated over all blood type combinations; here, Figure 10 displays that same information in tabular format, where "zone" numbers correspond to the CPRA zones sorted from least to greatest

Zone	RF	Iso	RBF	Poly	MLP	GP
1	3.18	2.86	2.74	2.18	2.84	3.02
2	3.02	2.37	3.25	2.24	3.04	3.25
3	3.29	2.90	3.37	3.35	4.38	3.97
4	11.01	7.90	28.89	17.69	20.7	8.02
Avg	5.12	4.01	9.56	6.36	7.74	4.57

Figure 10: A tabular display of Fig. 2. Here, "RBF" and "Poly" denote SVR with the RBF and polynomial kernels respectively.

Figure 3 in the main paper explores the specific case of an O-type donor and AB-type patient. Figure 11 shows the distribution on the estimated function mapping patient CPRA for that blood type combination O-AB in weeks, the acquisition function (in red) in arbitrary units, and the point at which the acquisition is optimized (the solid red line going down). We see that uncertainty is quite low, even after only a small number of (expensive) black box function queries. Figure 11 and 12 demonstrate that while all models had relatively similar performance, for patients with very high CPRA, only our model, random forest regression, and isotonic regression capture the final upward trend towards 250, the maximum number of match iterations. Figure 13 gives that same comparison for the AB-type donor, O-type patient class of vertex.

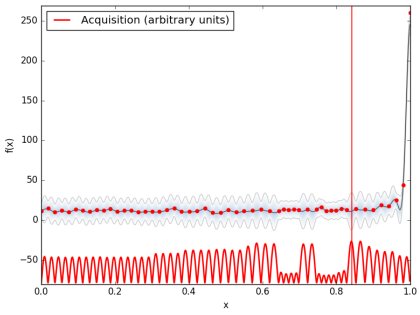

Figure 11: Gaussian process for O-type paired donor and AB-type patient (in weeks)

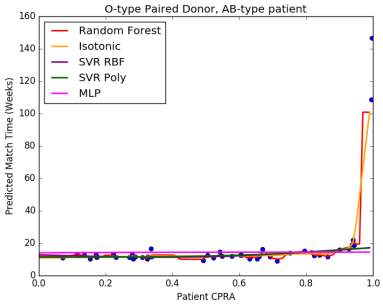

Figure 12: Prediction of expected waiting time (in weeks) by passively learned models for O-type paired donor and AB-type patient

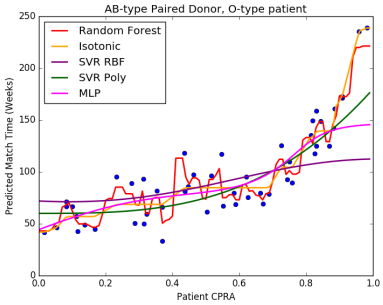

Figure 13: Prediction of expected waiting time (in weeks) by passively learned models for AB-type paired donor and O-type patient