Electronic Companion

EC.1. Additional Discussion of Related Learning Paradigms

EC.1.1. Inverse reinforcement learning

Inverse Reinforcement Learning (IRL) is the problem of estimating the reward function of a reinforcement learning problem (see Russell (1998), Ng and Russell (2000) and Arora and Doshi (2021) for a recent survey). Since reinforcement learning can be used to solve MDPs, early IRL methods are fundamentally similar to Inverse MDPs (see Section 3.3). However, modern IRL methods draw more from the machine learning and reinforcement learning literature (Sutton and Barto 2018). Nonetheless, the similarity of problems suggest that new data-driven inverse optimization methods may be obtained by adapting IRL techniques.

A reinforcement learning problem is defined by the tuple $(\mathcal{S}, \mathcal{A}, p, \boldsymbol{\theta}, \gamma)$ comprising a state space, action set, transition probabilities, reward function, and discount factor. We deviate slightly from the notation of Section 3.3 and refer to specific rewards and value functions as $\theta(s,a)$ and $v(s)$, respectively. We assume without loss of generality that there is an initial state s_0 from which all trajectories begin. In IRL, we may observe a policy $\hat{\boldsymbol{\pi}}(s)$ from an agent, or instead only observe N length T state-action trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^N$ where $\tau_i := \langle (s_0, a_0^i), (s_1^i, a_1^i), \ldots, (s_T^i, a_T^i)\rangle$. The goal is to estimate a reward vector $\boldsymbol{\theta}$ for which the observed policy or trajectories are optimal.

Early IRL methods observe a policy $\hat{\boldsymbol{\pi}}$ and solve an inverse problem with convex programming. For instance, the original Max-Margin framework estimates a reward function such that the actions taken by $\hat{\boldsymbol{\pi}}$ achieve higher expected rewards than any other actions (Russell 1998, Ng and Russell 2000, Abbeel and Ng 2004, Ratliff et al. 2006). Let $q(s,a) := \theta(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v(s')$ be the state-action q-function. Then, given a policy $\hat{\boldsymbol{\pi}}$, the Max-Margin loss function is

$$\ell_{\mathrm{MM}}(\hat{\boldsymbol{\pi}}) := \sum_{s \in \mathcal{S}} \left(q(s, \hat{\boldsymbol{\pi}}(s)) - \max_{a \in \mathcal{A} \setminus \{\hat{\boldsymbol{\pi}}(s)\}} q(s,a) \right).$$

The above loss computes the difference in q-values between the actions from an observed policy and the next best action. For MDPs where the reward only depends on the state, i.e., $\theta(s,a) = \theta(s)$ for all $a \in \mathcal{A}$, Ng and Russell (2000) note that this loss is a convex function of $\boldsymbol{\theta}$. They formulate a convex program where the estimated reward is constrained to ensure that the observed policy satisfies an optimality condition for the MDP. Below, we present a variant of the original formulation:

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}, v, q} \quad & \ell_{\mathrm{MM}}(\hat{\boldsymbol{\pi}}) - \kappa \|\boldsymbol{\theta}\|_1 \\ \text{s.t.} \quad & q(s,a) = \theta(s) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v(s'), \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \\ & v(s) \geq q(s,a), \quad \forall s \in \mathcal{S},\ a \in \mathcal{A} \\ & v(s) = q(s,a'), \quad \forall s \in \mathcal{S},\ a' = \hat{\boldsymbol{\pi}}(s). \end{aligned} \tag{EC.1}$$

Algorithm 1 General max-margin inverse reinforcement learning framework

Input: Data set of trajectories $\{\tau_i\}_{i=1}^N$

Output: Reward function estimate $\hat{\theta}(s,a) = \sum_{b=1}^B \hat{\theta}_b f^{(b)}(s,a)$

1: Initialize a reward estimate $\tilde{\boldsymbol{\theta}}$; Memory of rewards $\mathcal{P} = \emptyset$

2: **while** Convergence criteria not met **do**

3: Solve the reinforcement learning problem with $\tilde{\boldsymbol{\theta}}$ to obtain a candidate policy

$$\tilde{\boldsymbol{\pi}} \leftarrow \mathrm{RL}(\mathcal{S}, \mathcal{A}, p, \tilde{\boldsymbol{\theta}}, \gamma)$$

4: Update $\mathcal{P} \leftarrow \mathcal{P} \cup \{\tilde{\boldsymbol{\pi}}\}$ and solve an approximate Max-Margin IRL problem, e.g.,

$$\tilde{\boldsymbol{\theta}} \leftarrow \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\min} \quad \sum_{\tilde{\boldsymbol{\pi}} \in \mathcal{P}} \ell\left(v^{\tilde{\boldsymbol{\pi}}}(s_0),\ \frac{1}{N}\sum_{i=1}^N \left(\theta(s_0) + \sum_{t=1}^T \gamma^t \theta\left(s_t^{(i)}\right)\right)\right)$$

5: **return** Final reward estimate $\tilde{\boldsymbol{\theta}}$

The constraints of Problem (EC.1) match (9b)–(9c) but with the q-function. Ng and Russell (2000) further use a regularization term $\kappa \|\boldsymbol{\theta}\|_1$ that encourages "simpler" small-magnitude rewards.

Problem (EC.1) can be typically too difficult to solve for practical applications where the state and action spaces are large and we only observe trajectories rather than policies and state transition probabilities. The Max-Margin literature proposes two resolutions to address these concerns. First, we may model the reward as a linear combination of fixed basis functions $\theta(s) = \sum_{b=1}^B \theta_b f^{(b)}(s)$; this can reduce the number of variables to estimate and further ensure linearity in the parameters identical to convex-separable bases in the inverse optimization literature. Second, we may eschew designing a reward function itself and instead minimize the margin between the value function of the optimal policy $\boldsymbol{\pi}(\boldsymbol{\theta})$ and the empirical expected state-action q-function value, i.e.,

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \quad \ell\left(v^{\boldsymbol{\pi}(\boldsymbol{\theta})}(s_0),\ \frac{1}{N}\sum_{i=1}^N \left(\theta(s_0) + \sum_{t=1}^T \gamma^t \theta\left(s_t^{(i)}\right)\right)\right)$$

where $\{\langle (s_0, a_0^i), (s_1^i, a_1^i), \ldots, s_T^i, a_T^i)\rangle\}_{i=1}^N$ is the data set of observed trajectories and $\ell(v_1, v_2)$ is a penalty function on the difference of values. This problem is typically approximately solved by iteratively computing the margin with respect to a set of candidate policies $\tilde{\boldsymbol{\pi}}$. Algorithm 1 highlights the general steps of a Max-Margin method.

Note that the problem formulation of large-scale Max-Margin methods parallel the formulation of data-driven inverse optimization. In inverse optimization, inverse-feasibility becomes hard to satisfy with large decision data sets, necessitating data-driven loss functions that penalize the

violation of optimality conditions. On the other hand, classical IRL is too large to model using optimality criteria, leading to loss functions that penalize the sub-optimality of the trajectories.

Recent IRL methods relate more closely with modern machine learning, e.g., Maximum Entropy (Ziebart et al. 2008, Boularias et al. 2011), Bayesian IRL (Ramachandran and Amir 2007, Lopes et al. 2009, Levine et al. 2011), or supervised learning of q-values (Taskar et al. 2005). For example, the Maximum Entropy literature posits that in an MDP with stochastic state dynamics, the observed trajectories τ are drawn from an optimal policy and must have the highest likelihood over all other trajectories. This can recast IRL as a maximum likelihood problem $\max_\theta \mathbb{E}_{\mathbb{P}}[\log p_\theta(\tau)]$ where $p_\theta(\tau) \propto p(s_0)\prod_{t=0}^{T} p(s_{t+1}|s_t,a_t)\exp(\gamma^t\theta(s_t,a_t))$ follows a Boltzmann distribution.

We conclude this section by highlighting a common weakness of both inverse optimization and IRL. Both problems face the risk of obtaining uninformative degenerate estimates. Note that for any MDP, the all-zero reward function $\theta(s,a)=0$ will ensure that every policy (including an observed one) is optimal. In IRL, this necessitates well-designed objectives such as the Max-Margin. However, Ng et al. (1999) further show that the optimal policy for an MDP with reward function $\theta(s,a,s')$ is invariant to the transformation $\theta(s,a,s')+\gamma\Phi(s')-\Phi(s)$ for any *arbitrary* function $\Phi:\mathcal{S}\to\mathbb{R}$. Effectively, if a policy is optimal, there are an infinite number of reward functions for which it is so. To mitigate this problem, Fu et al. (2018) propose an adversarial IRL framework that estimates 'disentangled' rewards. Although not to the same extent, the scaling invariance property (see Section 4.2.2) is a similar characteristic in inverse optimization. However, scaling invariance is often easily mitigated by a normalization constraint in inverse optimization.

EC.1.2. Decision-aware learning

Consider a contextual optimization problem

$$\min_{\mathbf{x}} \left\{ f(\boldsymbol{\theta},\mathbf{x}) \,|\, \mathbf{x}\in\mathcal{X}(\boldsymbol{\theta}) \right\},$$

where $\boldsymbol{\theta}$ is unknown and must be predicted using contextual information $\mathbf{u}$. Given a historical data set $\{(\hat{\mathbf{u}}_i,\hat{\boldsymbol{\theta}}_i)\}_{i=1,\dots,N}$ of past contextual information (i.e., features) and observed values, a standard machine learning approach is to fit a predictive model to minimize the prediction error on $\boldsymbol{\theta}$. In contrast, decision-aware learning fits a predictive model that minimizes the *suboptimality* of the optimal decision $\mathbf{x}\in\mathcal{X}$ generated using the predicted value $\boldsymbol{\theta}$.

In this section, we formalize the decision-aware learning framework and highlight its connection with inverse optimization. To do so, we first consider a setting where we seek to fit a linear predictive model $\boldsymbol{\beta}^\top\mathbf{u}$ for $\boldsymbol{\theta}$ in a contextual optimization problem with a strictly convex objective function

$f(\boldsymbol{\theta}, \mathbf{x})$ and a convex feasible region $\mathcal{X}$. For each $\hat{\boldsymbol{\theta}}_i$, let $\hat{\mathbf{x}}_i$ denote the unique optimal solution to the corresponding contextual problem. The decision-aware learning problem is then modeled as

$$\mathbf{DAL}(\{(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i, \hat{\boldsymbol{\theta}}_i)\}_{i=1}^N) := \min_{\boldsymbol{\theta}_i, \mathbf{x}_i, \boldsymbol{\beta}} \quad \sum_{i=1}^N \left(f(\hat{\boldsymbol{\theta}}_i, \hat{\mathbf{x}}_i) - f(\hat{\boldsymbol{\theta}}_i, \mathbf{x}_i)\right) \tag{EC.2a}$$

$$\text{s.t.} \quad \mathbf{x}_i = \operatorname{argmin}\Big\{f(\boldsymbol{\theta}_i, \mathbf{x}) \,|\, \mathbf{x} \in \mathcal{X}\Big\}, \quad \forall i \in \{1, \ldots, N\} \tag{EC.2b}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\beta}^\top \hat{\mathbf{u}}_i, \quad \forall i \in \{1, \ldots, N\}. \tag{EC.2c}$$

This problem closely resembles the data-driven inverse optimization models covered in Section 4. Specifically, we could recast the decision-aware learning problem as an inverse problem where (i) the input data is observed decisions $\{\hat{\mathbf{x}}_i\}_{i=1}^N$ and contextual information $\{\hat{\mathbf{u}}_i\}_{i=1}^N$, (ii) the loss function is a variant of the absolute sub-optimality loss defined in Section 4.2, and (iii) each forward model instance has a context-dependent objective function defined by the linear function $\boldsymbol{\beta}^\top \mathbf{u}$.

The connection between decision-aware learning and inverse optimization can be extended beyond our illustrative example to more general problem structures, such as linear programs (Sun et al. 2023). More importantly, the similarities between the two problem classes imply that solution techniques developed for one area can potentially be relevant for the other. For example, solution methods for decision-aware learning revolve almost entirely around reformulating or approximating the problem so that iterative gradient-based methods can be applied, the latter of which are grounded in the machine learning literature (Sadana et al. 2023). On the other hand, inverse optimization models have typically been examined using ideas from the bilevel optimization community, where reformulation techniques and algorithms (e.g., cutting planes) can leverage specific problem structure and/or notions of optimality and duality (e.g., KKT conditions). In theory, both types of approaches can be tailored to both problem classes, but may lead to differences in computational performance and approximation quality (e.g., Jeong et al. 2022). We believe that there exists significant value in the cross-pollination of methodology between these two research areas (e.g., Muñoz et al. 2022, Sun et al. 2023).

EC.2. Additional Details for Select Applications

In this section we provide additional mathematical details of some applications that are discussed in the main body of the paper.

EC.2.1. Toll design for risk mitigation

Let $\mathcal{G}(\mathcal{V}, \mathcal{A})$ denote a network with nodes $\mathcal{V}$ and arcs $\mathcal{A}$. Assume the carrier's problem is a multi-commodity flow problem over $\mathcal{G}$ minimizing the cost of transporting S types of hazardous materials between a set of source-sink pairs. For each material type $s \in \{1, \ldots, S\}$, let c_a^s denote the cost of

transporting the material s on arc $a \in \mathcal{A}$ and let $x_a^s \in \{0,1\}$ be a binary variable that equals one if material s is transported on arc a. Given resource and flow balance constraints in the form of $\mathcal{X}$, the carrier's multi-commodity flow problem is

$$\min_{\mathbf{x}} \left\{ \sum_{s=1}^{S} \sum_{a \in \mathcal{A}} c_a^s x_a^s \;\middle|\; \mathbf{x} \in \mathcal{X} \right\}.$$

We use $\mathcal{X}^{\text{opt}}(\mathbf{c})$ to denote the set of optimal solutions to this problem under transportation costs $\mathbf{c}$. Now suppose the estimated risk posed by transporting material s over arc a is ρ_a^s. To find a set of arc tolls $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ that makes the minimum-risk solution optimal for the carrier, we can solve

$$\min_{\mathbf{x},\boldsymbol{\theta}} \left\{ \sum_{s=1}^{S} \sum_{a \in \mathcal{A}} \rho_a^s x_a^s \;\middle|\; \mathbf{x} \in \mathcal{X}^{\text{opt}}(\mathbf{c}+\boldsymbol{\theta}),\, \boldsymbol{\theta} \geq \mathbf{0} \right\}.$$

To solve this problem, we can use the master-subproblem technique, described in Section 6, which decouples the computation of the lowest-risk solution and the corresponding tolls into two sequential linear programs. Specifically, we first solve the master problem

$$\min_{\mathbf{x}} \left\{ \sum_{s=1}^{S} \sum_{a \in \mathcal{A}} \rho_a^s x_a^s \;\middle|\; \mathbf{x} \in \mathcal{X} \right\},$$

to yield a feasible flow $\hat{\mathbf{x}}$ that minimizes the risk. Then we solve an inverse optimization subproblem to obtain the tolls for which $\hat{\mathbf{x}}$ is now also optimal for the carrier, i.e.,

$$\min_{\boldsymbol{\theta}} \left\{ \sum_{s=1}^{S} \sum_{a \in \mathcal{A}} \theta_a^s \hat{x}_a^s \;\middle|\; \hat{\mathbf{x}} \in \mathcal{X}^{\text{opt}}(\mathbf{c}+\boldsymbol{\theta}),\, \boldsymbol{\theta} \geq \mathbf{0} \right\}.$$

Note that this inverse problem computes the *minimum* amount of tolls that must be collected from the carrier to induce the minimum-risk solution.

EC.2.2. Estimating traffic equilibrium models

We first describe the traffic assignment model (i.e., the forward problem which produces equilibrium conditions), then provide details on inverse optimization framework used to estimate this model.

The forward problem. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ define a road network consisting of a set of m nodes $\mathcal{V}$ and a set of n directed arcs $\mathcal{A}$, and let $\mathcal{P} = \mathcal{V} \times \mathcal{V}$ denote the set of all node pairs in the network. Let $\mathbf{A} \in \{0,1,-1\}^{m \times n}$ denote the node-arc incidence matrix. The value $d_{i,j}$ denotes the demand between each pair of nodes $(i,j) \in \mathcal{P}$, and we define a corresponding vector $\mathbf{b}^{i,j} \in \mathbb{R}^m$ where $\mathbf{b}^{i,j}$ is a vector of all zeros except for elements i and j where $b_i^{i,j} = d_{i,j}$ and $b_j^{i,j} = -d_{i,j}$. Let $\mathbf{x}^{i,j} = (x_1^{i,j}, \ldots, x_n^{i,j})$ denote a vector of flows on each arc corresponding to demand between node i and j. Finally, let $\mathcal{X}$ be the set of aggregate flows satisfying all pairwise demand, i.e.,

$$\mathcal{X} = \left\{ \mathbf{x} \;\middle|\; \mathbf{x} = \sum_{(i,j) \in \mathcal{P}} \mathbf{x}^{i,j},\; \mathbf{A}\mathbf{x}^{i,j} = \mathbf{b}^{i,j},\; \mathbf{x}^{i,j} \geq \mathbf{0}, \quad \forall (i,j) \in \mathcal{P} \right\}.$$

For a given flow $\mathbf{x} \in \mathcal{X}$, let $\mathbf{f}(\mathbf{x}) = (f_1(x_1), \ldots, f_n(x_n))$ be the vector of marginal travel time costs on each arc under $\mathbf{x}$. In the transportation literature, these costs are commonly modeled as

$$f_a(x_a) = c_a\, g\left(\frac{x_a}{m_a}\right),$$

where c_a is a parameter measuring the uncongested travel time of arc a, m_a is a parameter corresponding to the "capacity" of an arc (e.g., the number of lanes and the speed limit) and $g(x)$ is a monotonically increasing function modeling congestion effects. Thus, arc costs are differentiated only by c_a and m_a, where they increase when c_a or x_a increase, or when m_a decreases.

Given a demand matrix and known cost functions, different flow solutions can be computed using various transportation models which differ by the underlying assumptions made. A widely studied model is the traffic assignment problem (Dafermos and Sparrow 1969, Patriksson 2015), which produces solutions describing decentralized traffic flow, where every driver has complete information on costs and makes their routing decisions independent of decisions made by other drivers. This defines the forward model

$$\min_{\mathbf{x}} \left\{ \sum_{a=1}^{n} \int_0^{x_a} f_a(s)\, ds \;\middle|\; \mathbf{x} \in \mathcal{X} \right\}. \tag{EC.3}$$

Since $\mathbf{f}(\mathbf{x})$ represents the marginal travel time cost, it is the gradient of the objective function in the forward model. This model is considered to be a good approximation for describing real-world traffic movement. However, for this model to be of practical use, $\mathbf{f}(\mathbf{x})$, and more specifically the $g(\cdot)$ function, must be known. The standard approach in transportation modeling is to assume a specific $g(\cdot)$ function, a common choice being $g(\frac{x_a}{m_a}) = (1 + 1.15(\frac{x_a}{m_a})^4)$, as well as a specific value of m_a (Chow et al. 2014, Bertsimas et al. 2015). However, assuming such functions, rather than deriving from data, can result in poor modeling and out-of-sample performance.

The inverse problem. Suppose we observe spatiotemporal data of transportation flows and demands over T periods, denoted by $\{(\hat{\mathbf{x}}_t, \mathcal{X}_t)\}_{t=1}^{T}$ with $\hat{\mathbf{x}}_t \in \mathcal{X}_t$ for all t. In the inverse problem, we estimate the marginal travel time cost functions $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ where $f_a(x_a, \boldsymbol{\theta}) = c_a g(\frac{x_a}{m_a}, \boldsymbol{\theta})$ are now parametrized by $\boldsymbol{\theta}$. Following Bertsimas et al. (2015) and Zhang et al. (2018), we model $g(\frac{x_a}{m_a}, \boldsymbol{\theta})$ using polynomial kernels

$$g\left(\frac{x_a}{m_a}, \boldsymbol{\theta}\right) = 1 + \theta_1 \left(\frac{x_a}{m_a}\right) + \ldots + \theta_n \left(\frac{x_a}{m_a}\right)^n. \tag{EC.4}$$

In order to estimate $\boldsymbol{\theta}$ using the data set $\{(\hat{\mathbf{x}}_t, \mathcal{X}_t)\}_{t=1}^{T}$, Bertsimas et al. (2015) leverage an important property from the transportation literature, that when $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ is strongly monotonic and continuously differentiable, the optimal traffic flows satisfy a Wardrop equilibrium (Dafermos and Sparrow

1969, Patriksson 2015). That is, the unique optimal solution $\mathbf{x}^*$ to (EC.3) is also the unique solution to the following set of variational inequalities

$$\mathbf{f}(\mathbf{x}^*, \boldsymbol{\theta})^\top (\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}.$$

The Wardrop equilibrium ensures that for every $(i,j) \in \mathcal{P}$, and for any route between the pair with positive flow in $\mathbf{x}^*$, the cost of traveling along that route is no greater than the cost of traveling along any other feasible routes between i and j (Patriksson 2015). Practically, this implies that every driver acts selfishly and takes the lowest cost route.

With this connection to variational inequalities, and with the observation that the forward problem is conic, we can employ the Inverse Variational Inequality problem (see Section 4.3, and in particular Theorem 6) for problem (EC.3) to estimate the arc cost functions by solving the following inverse optimization problem

$$\begin{aligned}
\min_{\boldsymbol{\epsilon}, \boldsymbol{\lambda}, \boldsymbol{\theta}} \quad & \|\boldsymbol{\epsilon}\|_p + \kappa \sum_{i=1}^{n} \delta_i \theta_i^2 \\
\text{s.t.} \quad & \sum_{a \in \mathcal{A}} c_a \hat{x}_{a,t}\, g\left(\frac{\hat{x}_{a,t}}{m_a}, \boldsymbol{\theta}\right) - \sum_{(i,j) \in \mathcal{P}} (\mathbf{b}_t^{ij})^\top \boldsymbol{\lambda}_t^{ij} \leq \epsilon_t, \quad \forall t \in \{1, \ldots, T\}, \\
& \mathbf{A}_a^\top \boldsymbol{\lambda}_t^{ij} - c_a\, g\left(\frac{\hat{x}_{a,t}}{m_a}, \boldsymbol{\theta}\right) \leq 0, \quad \forall a \in \mathcal{A},\, t \in \{1, \ldots, T\}, \\
& \boldsymbol{\theta} \in \boldsymbol{\Theta}.
\end{aligned}$$

The vector $\mathbf{A}_a$ denotes the a-th row of matrix $\mathbf{A}$, and the variables $\boldsymbol{\lambda}_t^{ij} \in \mathbb{R}^m$ are duals associated with data point t and node-pair $(i,j) \in \mathcal{P}$. The parameters δ_i define a penalization term for polynomial kernels (see Bertsimas et al. 2015, Zhang et al. 2018). The set $\boldsymbol{\Theta}$ includes constraints that impose monotonicity on $\mathbf{f}(\cdot)$ over the observed flow ranges, either in the form of a general constraint $\boldsymbol{\theta} \geq \mathbf{0}$ or as a set of individual constraints on each arc in the form of $f_a(x_a) \geq f_a(\tilde{x}_a)$ for all observed arc flows x_a and $\tilde{x}_a$ where $x_a \geq \tilde{x}_a$. Note that because $g(\frac{x_a}{m_a}, \mathbf{0}) = 1$ in equation (EC.4), we avoid inferring the "trivial" objective vector $g(\cdot) = 0$ that would render all observed solutions equivalent and optimal. Finally, the regularization term κ in the inverse problem can be tuned using cross-validation.

Zhang et al. (2016) and Zhang et al. (2018) show that the estimated cost functions can be used in a different model to measure network efficiency. Specifically, we can use $\{f_a(\cdot)\}_{a \in \mathcal{A}}$ to solve a system-optimal routing model, which minimizes the total cumulative cost of travel for all drivers:

$$\min_{\mathbf{x}} \left\{ \sum_{a \in \mathcal{A}} x_a f_a(x_a) \,\middle|\, \mathbf{x} \in \mathcal{X} \right\}. \tag{EC.6}$$

Unlike solutions from model (EC.3), solutions from model (EC.6) are Pareto efficient. The ratio of the total travel time between system-optimal and user-optimal solutions, known as the "price

of anarchy", provides a measure of the efficiency of the system. While the ratio has been studied extensively as a theoretical concept, the inverse optimization approach enable the authors to provide one of the first empirical estimates of this ratio.

EC.2.3. Inferring constraints of market-clearing models

Consider a market with J participants situated across I different nodes. Let v_j denote the total amount of electricity consumed ($v_j < 0$) or produced ($v_j > 0$) by participant $j \in \{1, \ldots, J\}$ who is situated at node $n(j)$. Let $S_j(v_j)$ describe an offer or bid function submitted by participant j, i.e., $S_j(v_j)$ (or $-S_j(v_j)$) denotes the price that a consumer (or producer) j is willing to pay (or be paid) for v_j. We assume $S_j(v_j)$ is stepwise linear. Finally, let x_i represent the total amount of electricity consumed or produced at node $i \in \{1, \ldots, I\}$. The market-clearing model as defined in Birge et al. (2017) maximizes social welfare subject to constraints on M different transmission links as well as constraints $\mathbf{v} \in \mathcal{V}$ defining individual production and consumption, i.e.,

$$\max_{\mathbf{x},\mathbf{v}} \quad \sum_{j=1}^{J} S_j(v_j) \tag{EC.7a}$$

$$\text{s.t.} \quad \sum_{j:n(j)=i} v_j = x_i, \quad \forall i \in \{1, \ldots, I\}, \tag{EC.7b}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \tag{EC.7c}$$

$$\mathbf{v} \in \mathcal{V}. \tag{EC.7d}$$

The matrix $\mathbf{A} \in \mathbb{R}^{M \times I}$ and vector $\mathbf{b} \in \mathbb{R}^M$ define the grid and the transmission capacities (based on DC power flow). The vector of optimal dual variables $\hat{\boldsymbol{\pi}} \in \mathbb{R}^I$ and $\hat{\boldsymbol{\lambda}} \in \mathbb{R}^M$ corresponding to constraints (EC.7b) and (EC.7c) represent the prices of electricity at every node and the shadow prices of various capacity constraints, respectively; these are used to decide payments.

To recover the matrix $\mathbf{A}$, Birge et al. (2017) assume a noise-free environment and use a publicly available data set $\{(\hat{\mathbf{x}}_i, \hat{\boldsymbol{\lambda}}_i, \hat{\boldsymbol{\pi}}_i)\}_{i=1}^N$ where $N \geq M$ to find a matrix $\mathbf{A}$ and vectors $\mathbf{b}_i$ that satisfy the following set of optimality conditions (Birge et al. (2017) include other constraints related to the physical transmission of electricity, which we omit here for simplicity):

$$\boldsymbol{\pi}_i = \hat{\boldsymbol{\lambda}}_i \mathbf{A}, \ \mathbf{A}\hat{\mathbf{x}}_i \leq \mathbf{b}_i, \ \hat{\boldsymbol{\lambda}}_i \odot (\mathbf{A}\hat{\mathbf{x}}_i - \mathbf{b}_i) = \mathbf{0}, \quad \forall i \in \{1, \ldots, N\}.$$

Finally, we note that the operations of electricity markets vary significantly across geographic regions, and these distinctions offer new research opportunities. For example, Ruiz et al. (2013) examine a market where the model constraints are published rather than the bids. The authors describe a duality-based inverse optimization model to infer stepwise bid functions of producers.

EC.2.4. Clinical pathway concordance measurement

Let $\mathcal{G}$ denote a graph with nodes $\mathcal{N}$ and arcs $\mathcal{A}$. The nodes describe the set of activities patients can undertake, including concordant activities like medical imaging and treatment, and discordant activities like emergency department visits and extra consultations. A walk through the graph accumulates costs along the arcs it traverses. Clinical pathways developed by experts are assumed to be shortest paths through $\mathcal{G}$. Implicit costs of arcs between activities can then be estimated with model (26), which identifies a single set of arc costs $\boldsymbol{\theta}$ that minimizes the aggregate sub-optimality of clinical pathways with respect to the difference between their costs and the shortest path cost.

While model (26) considers the expert-defined clinical pathways as input, it does not consider actual patient data from either successful or unsuccessful clinical workflows. Consequently, Chan et al. (2022) employ a second-stage inverse problem to refine the cost vector $\boldsymbol{\theta}^*$ from model (26) with real patient pathways. Consider a set of patient-traversed pathways from patients who survived their cancer, $\hat{\mathbf{x}}_1^s, \ldots, \hat{\mathbf{x}}_S^s$ and a set of patient-traverse pathways from patients who died, $\hat{\mathbf{x}}_1^d, \ldots, \hat{\mathbf{x}}_D^d$. The refined problem penalizes the duality gaps with respect to patients who survived and encourages higher duality gaps for patients who died, while fixing the optimal duality gaps ϵ_i^{r*} from problem (26). We can write this problem as

$$\begin{aligned}
\min_{\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\epsilon}^s,\boldsymbol{\epsilon}^d} \quad & \frac{D}{S}\sum_{j=1}^{S}\epsilon_j^s - \sum_{k=1}^{D}\epsilon_k^d \\
\text{s.t.} \quad & \mathbf{A}^\top\boldsymbol{\lambda} \le \boldsymbol{\theta} \\
& \boldsymbol{\theta}^\top\hat{\mathbf{x}}_i^r = \mathbf{b}^\top\boldsymbol{\lambda} + \epsilon_i^{r*}, \quad \forall i \in \{1,\ldots,N\} \\
& \boldsymbol{\theta}^\top\hat{\mathbf{x}}_j^s = \mathbf{b}^\top\boldsymbol{\lambda} + \epsilon_j^{s}, \quad \forall j \in \{1,\ldots,S\} \\
& \boldsymbol{\theta}^\top\hat{\mathbf{x}}_k^d = \mathbf{b}^\top\boldsymbol{\lambda} + \epsilon_k^{d}, \quad \forall k \in \{1,\ldots,D\} \\
& \|\boldsymbol{\theta}\|_\infty = 1 \\
& \mathbf{A}\boldsymbol{\theta} = \mathbf{0}.
\end{aligned} \tag{EC.8}$$

The objective minimizes (maximizes) the aggregate sub-optimality with respect to the patients who survived (died). Because ϵ_q^{r*} is fixed, this model chooses among the optimal cost vectors from model (26) to find one that maximizes the "separation" between the pathway costs of patients who survived from those who died.

Finally, using the optimal cost vector $\boldsymbol{\theta}^*$ from problem (EC.8), Chan et al. (2022) construct a concordance metric $\omega \in [0,1]$ as follows:

$$\omega(\hat{\mathbf{x}}) = 1 - \frac{\boldsymbol{\theta}^{*\top}\hat{\mathbf{x}} - \boldsymbol{\theta}^{*\top}\mathbf{x}^*}{M(\hat{\mathbf{x}}) - \boldsymbol{\theta}^{*\top}\hat{\mathbf{x}}}.$$

This metric measures the cost difference between a patient pathway $\hat{\mathbf{x}}$ and a shortest path $\mathbf{x}^*$, and normalizes it based on the cost difference between the longest walk with the same number steps

as $\hat{\mathbf{x}}$ (denoted $M(\hat{\mathbf{x}})$) and a shortest path. Using ω, it becomes possible to rigorously score any patient pathway $\hat{\mathbf{x}}$ against the clinical pathways. Using a real dataset of colon cancer patients, the authors establish a statistically significant association between concordance and mortality, even after adjusting for patient covariates, which supports the clinical meaningfulness of the metric.

An extension that has not been considered yet is the incorporate of cost of care into the concordance measurement. This can be done by modifying formulation (EC.8) to favor patient pathways that are lower cost (in terms of real dollar amounts of imaging, treatment, diagnostic tests, etc.), rather than separating patients by their survival outcome.

EC.2.5. Radiation therapy treatment planning

The optimization problem contains multiple objectives used to balance multiple (potentially dozens) conflicting objectives, such as escalating dose to the tumor while minimizing dose to the healthy organs. The overall objective is formed by taking a weighted combination of the various objectives $f(\mathbf{x}) = \sum_{k=1}^{K} \theta_k f_k(\mathbf{x})$ where θ_k is the weight of the k-th objective $f_k(\mathbf{x})$ for $k \in \{1, \ldots, K\}$. Assuming linear objectives (i.e., $f_j(\mathbf{x}) = \mathbf{c}^{j\top}\mathbf{x}$), the objective function can be rewritten as $\boldsymbol{\theta}^\top \mathbf{C}\mathbf{x}$, where the j-th row of $\mathbf{C}$ is $\mathbf{c}^j$.

The traditional clinical procedure of designing treatments involves manually selecting the objective function weights, solving the optimization problem, evaluating the treatment using several quantitative and qualitative metrics, and then iterating, if needed. As an alternative, inverse optimization can infer appropriate weights from historical treatments (Chan et al. 2014). We present a simplified version of this model below.

Let $\mathcal{B}$ be the sent of beamlets and x_b be the intensity of beamlet b. Let $\mathcal{T}$ be the set of voxels (volumetric pixels) in the tumor and $\mathcal{V}$ be the set of all voxels. Let $\mathcal{O}_k$ be the set of healthy voxels corresponding to the k-th objective. Each objective $f_k(\mathbf{x})$ penalizes the total dose delivered above a threshold τ_v^k for each voxel v, i.e.,

$$f_k(\mathbf{x}) := \sum_{v \in \mathcal{O}_k} \max\left\{0, \sum_{b \in \mathcal{B}} D_{v,b} x_b - \tau_v^k\right\},$$

where $D_{v,b}$ is the dose deposited to voxel v by unit intensity of beamlet b. The complete formulation of the forward problem is

$$\begin{aligned}
\min_{\mathbf{x}} \quad & \sum_{k=1}^{K} \theta_k \sum_{v \in \mathcal{O}_k} \max\left\{0, \sum_{b \in \mathcal{B}} D_{v,b} x_b - \tau_v^k\right\} \\
\text{s.t.} \quad & \sum_{b \in \mathcal{B}} D_{v,b} x_b \geq l_v, \quad \forall v \in \mathcal{T}, \\
& \sum_{b \in \mathcal{B}} D_{v,b} x_b \leq u_v, \quad \forall v \in \mathcal{V}, \\
& \mathbf{x} \in \mathcal{X},
\end{aligned} \tag{EC.9}$$

where l_v and u_v denote lower and upper bound constraints on the tumor and healthy voxels, respectively, and $\mathcal{X}$ describes a set of linear constraints on the intensity values including nonnegativity.

Both the Absolute and Relative Sub-optimality loss functions (see Sections 4.2.1 and 4.2.2) have been used in the literature in the inverse formulation for the above forward problem.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*. ACM, 2004.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015.

John R Birge, Ali Hortaçsu, and J Michael Pavlin. Inverse optimization for the recovery of market structure from market outcomes: An application to the miso electricity market. *Operations Research*, 65(4): 837–855, 2017.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 182–189, 2011.

Timothy CY Chan, Tim Craig, Taewoo Lee, and Michael B Sharpe. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3):680–695, 2014.

Timothy CY Chan, Maria Eberg, Katharina Forster, Claire Holloway, Luciano Ieraci, Yusuf Shalaby, and Nasrin Yousefi. An inverse optimization approach to measuring clinical pathway concordance. *Management Science*, 68(3):1882–1903, 2022.

Joseph YJ Chow, Stephen G Ritchie, and Kyungsoo Jeong. Nonlinear inverse optimization for parameter estimation of commodity-vehicle-decoupled freight assignment. *Transportation Research Part E: Logistics and Transportation Review*, 67:71–91, 2014.

Stella C Dafermos and Frederick T Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards B*, 73(2):91–118, 1969.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Jihwan Jeong, Parth Jaggi, Andrew Butler, and Scott Sanner. An exact symbolic reduction of linear smart predict+ optimize to mixed integer linear programming. In *International Conference on Machine Learning*, pages 10053–10067. PMLR, 2022.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. *Advances in Neural Information Processing Systems*, 24:19–27, 2011.

Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer, 2009.

Miguel Angel Muñoz, Salvador Pineda, and Juan Miguel Morales. A bilevel framework for decision-making under uncertainty with contextual information. *Omega*, 108:102575, 2022.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pages 663–670, 2000.

Andrew Y Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pages 278–287, 1999.

Michael Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *International Conference on Machine Learning*, pages 729–736, 2006.

Carlos Ruiz, Antonio J Conejo, and Dimitris J Bertsimas. Revealing rival marginal offer prices via inverse optimization. *IEEE Transactions on Power Systems*, 28(3):3056–3064, 2013.

Stuart Russell. Learning agents for uncertain environments. In *Conference on Computational Learning Theory*, pages 101–103, 1998.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision making under uncertainty. *arXiv preprint arXiv:2306.10374*, 2023.

Chunlin Sun, Shang Liu, and Xiaocheng Li. Maximum optimality margin: A unified approach for contextual linear programming and inverse linear programming. *arXiv preprint arXiv:2301.11260*, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *International Conference on Machine Learning*, pages 896–903, 2005.

Jing Zhang, Sepideh Pourazarm, Christos G Cassandras, and Ioannis Ch Paschalidis. The price of anarchy in transportation networks by estimating user cost functions from actual traffic data. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 789–794. IEEE, 2016.

Jing Zhang, Sepideh Pourazarm, Christos G Cassandras, and Ioannis Ch Paschalidis. The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies. *Proceedings of the IEEE*, 106(4):538–553, 2018.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.