arXiv:2304.03367v3 [cs.RO] 4 Dec 2023

通过逆优化从专家示范中推断控制任务的约束

Dimitris Papadimitriou 和 Jingqi Li

摘要——**推断未知约束是许多机器人学应用中的一项具有挑战性且至关重要的问题。当只有专家示范可用时，要有效部署其他智能体，推断未知的定义域约束就变得至关重要。在本文中，我们提出一种方法，通过观察专家示范来推断控制任务中的仿射约束。我们将约束推断问题建模为一个逆优化问题，并提出一种交替优化方案，通过最小化 KKT残差目标函数来推断未知约束。我们通过一系列仿真展示了该方法的有效性，并表明与近期的基线方法相比，该方法能够推断出保守性较低的约束，同时保持相当的安全保证。**

I. 引言

为控制和强化学习（RL）任务指定奖励函数并非易事。一种指定此类奖励函数的方法是通过逆强化学习（IRL），利用专家示范推断未知的奖励函数 [1]。获得奖励函数后，便可以推导出策略。在约束环境中，当约束事先已知时，通过控制或强化学习来获得策略是相对容易处理的任务。在许多情况下，奖励函数的指定十分直观。例如，机器人可以将其到目标位置的距离倒数用作奖励函数。然而，为当前任务指定相应的约束并不总是那么直接。

处理未知约束下的策略学习问题，一种可行途径是使用模仿学习（IL）算法 [2]。然而，简单地模仿专家智能体存在一些弊端。首先，推断出的策略仅适用于示范任务，可能无法泛化到新任务。其次，在特定动力学模型下推断出的策略，并不能保证适用于具有不同状态转移动力学的智能体。另一方面，给定专家示范后，我们可以先推断约束，然后学习安全策略。强化学习中的约束推断问题大体上是一个新兴且活跃的研究领域，迄今已提出了许多方法 [3]。

控制应用中的约束推断是一个尚未得到广泛研究的重要问题 [4]。鉴于大多数控制任务都被建模为一个优化问题，因此可以利用逆优化理论中的相关要素来推断约束。在逆优化（IO）中，在给定.....的情况下推断优化问题的参数

作者谨此感谢 S.Sojoudi 教授提出的富有洞见的意见。作者来自加州大学伯克利分校。如有事宜，请联系 todimitri@berkeley.edu

获取其最优解 [5]。尽管大多数研究都围绕目标函数中的参数推断展开，这与强化学习中的奖励学习相对应，但近期研究开始关注约束相关参数的推断。

在这项工作中，我们提出了一种基于专家示范推断控制任务中约束的方法。我们的贡献可总结如下：*1)* 我们证明，当已知未知约束的确切数量以及约束处于绑定状态的状态时，可以从专家轨迹数据中精确恢复未知仿射约束；*2)* 当缺乏约束处于绑定状态的时间实例的先验知识时，我们通过最小化最优控制问题的 KKT 残差，提出了一种推断未知约束的方法；与依赖混合整数线性规划（MILP）的现有方法相比，该问题可以更高效地求解；*3)* 为应对不知道未知约束确切数量这一挑战，我们提出了一种贪心算法，该算法依次推断使KKT残差最小化的仿射约束。我们通过实验表明，我们的方法对观测噪声和次优示范具有鲁棒性，并且在具有相同级别安全保证的情况下，能够推断出比最先进的基线方法更不保守的约束。

本文其余部分安排如下。第二节回顾相关文献。第三节对约束推断问题进行表述。第四节介绍我们的主要结果。第五节通过仿真展示我们方法的性能。

II. 相关工作

本节概述强化学习和控制领域中现有的约束推断研究，并进一步介绍逆向优化文献中的相关工作，因为这些工作与我们的框架尤为相关。

约束逆强化学习 *(CIRL)*：在强化学习的背景下，[6] 提出了一种贪心方法，以在离散状态和动作空间中逐步添加约束。这是通过假设可以访问奖励函数来实现的，因此可以评估每个状态或动作受到约束的可能性。在每次迭代中，根据最大似然准则，将最有可能受到约束的状态或动作分类为约束。在 [7] 中提出了一个连续状态空间框架，作者在其中提出了一种学习约束分类器的方法。最后，在 [8] 中，作者提出了一种贝叶斯方法来推断约束。估计完整的约束后验分布还可以进一步量化估计不确定性。

控制中的约束推断：在机器人学的背景下， [9] 中的作者提出了一种方法，该方法基于机器人与物品交互的专家示范来推断约束。推断约束从参数化约束模型目录中选择，使得在这些约束下，机器人示范能够被复现。在 [10], 中，专家示范的单元占用情况会被计算，随后用于整数规划公式，以提供约束的估计值。约束也可以控制屏障函数的形式进行建模。 [11] 中的作者利用专家示范来学习具有可证明安全保证的控制屏障函数。

此外， [12], [13]中的作者推断用于在约束环境中获取控制策略的目标函数参数。然而，需要注意的是，未知参数仅存在于目标函数中，而不存在于约束中。优化问题的KKT 最优性条件也可用于推断约束集。在 [14], 中，在构建候选约束集之后，逆优化问题的拉格朗日乘子被用于推断约束。与我们框架最接近的工作是 [4],, 其中作者利用KKT条件构建状态-动作空间的安全区域和不安全区域，从而隐式地构建约束本身。尽管这种方法也可以推断目标函数中的不确定性，但我们选择仅专注于约束恢复任务。

逆优化：逆优化广泛指通过观察最优解 [5], [15]来推断优化问题参数的问题。大多数研究都围绕估计目标函数中的参数展开。对于线性目标函数和约束， [16] 将逆优化问题同样表述为线性问题。关于约束推断， [17] 推断线性前-向优化问题的线性约束左侧。最后， [18] 为推断线性优化问题中线性约束的左侧和右侧提供了一种凸形式。

在我们的工作中，我们基于最小化优化问题的 KKT 残差来设计约束推断方法。

KKT 残差与交替优化：在最优控制中，KKT 条件是状态与控制序列达到最优的必要条件。近期研究 [19], [20], [21]提出最小化 KKT 条件的残差，以找到最佳代价函数，使计算得到的轨迹与专家示范相匹配。KKT 残差表述的一个优点是，对于某些类别的问题，它关于未知参数可以是凸的 [19], [22]。此外，KKT 残差表述可以扩展到求解状态和控制上带有不等式约束的逆最优控制问题 [19]。

III. 问题表述

在这项工作中，我们旨在提供一种从约束最优控制问题中推断约束的方法。我们首先将经典的前向控制优化问题形式化如下。

定义 *1*： 考虑一个离散时间动力系统$x_{t+1} = f(x_t, u_t)$, 其中 $x_t \in \mathbb{R}^n$和 $u_t \in \mathbb{R}^m$分别表示时间 $t \in \{0, 1, 2, \dots\}$时的状态和控制输入。我们将从 t =0 到 $t = T$的控制和状态轨迹分别记为 $\mathbf{u} = \{u_0, \dots, u_{T-1}\}$和$\mathbf{x} = \{x_0, x_1, \dots, x_T\}$。所选的初始状态记为 x_0。对于具有二次代价函数 $c(x, u)$: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$、终端代价函数$c_T(x)$: $\mathbb{R}^n \to \mathbb{R}$ 以及约束函数 $g_\theta(x)$: $\mathbb{R}^n \to \mathbb{R}$的控制任务，其中这些函数由向量 $\theta \in \mathbb{R}^d$参数化，我们将正向控制问题**FCP**(θ)定义如下，

$$\begin{aligned} \textbf{FCP}(\theta) := \min_{\{u_t\}_{t=0}^{T-1}} & \sum_{t=0}^{T-1} c(x_t, u_t) + c_T(x_T) \\ \text{s.t. } & x_{t+1} = f(x_t, u_t), t = 0, \dots, T-1 \\ & g_\theta(x_t) \le 0, t = 0, \dots, T. \end{aligned} \tag{1}$$

在我们的工作中，我们聚焦于状态约束，但我们的方法也可以轻松用于状态-动作约束。已知来自 **FCP** 的最优解 $\mathbf{u}^* := \{u_t^*\}_{t=0}^{T-1}$，我们可以将逆控制问题（**ICP**）表述为一个优化问题，利用涉及该最优解的性能指标和约束来估计未知参数。

定义 *2*： 根据定义 1，我们将逆控制问题定义为 **ICP**($\mathbf{x}^*$,$\mathbf{u}^*$):= $\min_\theta \{\ell_{\mathbf{x}^*,\mathbf{u}^*}(\theta) | h_{\mathbf{x}^*,\mathbf{u}^*}(\theta) \le 0\}$ 通过适当选择损失函数 ℓ 和约束 h，恢复未知的约束参数 θ；其中$\mathbf{x}^*$,$\mathbf{u}^*$ 表示**FCP**的最优状态和输入解。

鉴于这些定义，我们希望解决以下问题。假设我们给定一组轨迹$\{\tau_i\}_{i=1}^N$，其中每个 $\tau_i = \{x_0, u_0, x_1, u_1, \dots, x_T\}$。在**ICP** 表述中，我们应使用什么目标函数 ℓ 和约束 h，才能高效地恢复未知约束参-ˆ数 θ？ 推断参数 θ 应满足：所诱导的最优轨迹应接近专家示范，并且满足未知约束。

假设 *1*：我们假设拥有一个专家数据集，其中包含若干条专家状态和控制输入轨迹，这些轨迹能够最优地求解**FCP**(θ)。

在本文余下部分，我们聚焦于线性动力学函数和有限时域任务。具体而言，我们考虑如下形式的线性动力学。

$$x_{t+1} = Ax_t + Bu_t, \tag{2}$$

其中 $x_t \in \mathbb{R}^n$ 和 $u_t \in \mathbb{R}^m$。此外， $A \in \mathbb{R}^{n \times n}$ 和 $B \in \mathbb{R}^{n \times m}$ 被视为已知矩阵。下文将详细阐述在具有线性约束的二次代价函数情况下的 **FCP** 和 **ICP** 。

A. 二次目标与线性约束

我们考虑具有有限时域 T、线性动力学、二次代价函数以及 M 个不同线性的控制任务

如下形式的约束

$$\begin{aligned}\min_{\{u_t\}_{t=0}^{t=T-1}} & \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t + x_T^\top Q x_T \\ \text{s.t. } & x_{t+1} = Ax_t + Bu_t, \forall t \in \{0,\dots,T-1\} \\ & c_i^\top x_t \le d_i, \forall t \in \{0,\dots,T\}, i \in \{1,\dots,M\},\end{aligned} \tag{3}$$

其中， Q 和 R 分别是与状态和控制输入相关的代价矩阵。鉴于我们研究的是线性约束，我们感兴趣并希望推断的未知参数 θ 是用于参数化第 i个约束的 $c_i \in \mathbb{R}^n$ 和 $d_i \in \mathbb{R}$ ，其中 $i \in \{1,2,\dots,M\}$。令 x_0为初始条件， $U = [u_0, u_1 \dots, u_{T-1}]^\top \in \mathbb{R}^{mT}$为控制输入序列的紧凑表示。该问题可简洁地重写为

$$\begin{aligned}\min_{U\in\mathbb{R}^{mT}} & (GU + Hx_0)^\top (I_T \otimes Q)(GU + Hx_0) \\ & + U^\top (I_T \otimes R) U \\ \text{s.t. } & C(GU + Hx_0) \le D,\end{aligned} \tag{4}$$

其中， $G \in \mathbb{R}^{nT\times mT}$、 $H \in \mathbb{R}^{nT\times n}$、 $C \in \mathbb{R}^{MT\times nT}$和 $D \in \mathbb{R}^{MT}$ 定义如下

$$G = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{T-1}B & A^{T-2}B & A^{T-3}B & \cdots & B \end{bmatrix}, H = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^T \end{bmatrix},$$

$$C = \begin{bmatrix} I_T \otimes c_1^\top \\ \vdots \\ I_T \otimes c_M^\top \end{bmatrix}, \quad D = \begin{bmatrix} \mathbf{1}_T d_1 \\ \vdots \\ \mathbf{1}_T d_M \end{bmatrix}, \tag{5}$$

其中， I_n 表示单位矩阵， $\mathbf{1}_n$ 表示全一向量，两者的大小均为 n。为简便起见，在矩阵定义中，我们用0表示适当维度的全零矩阵。

IV. 约束推断

本节介绍使用逆问题表述的约束恢复方法。首先，我们给出实现精确约束恢复的条件。随后，我们将推导一种更通用的方法，该方法基于 KKT 最优性条件进行约束推断。

A. 精确约束恢复

在某些假设下，可以通过求解线性方程组实现精确约束推断。当约束的确切数量以及约束处于绑定状态的状态均为先验已知时，就可以做到这一点。首先，我们给出在已知约束右端项 d_i、 $i = 1$、.....、 M的情况下，实现精确约束推断的条件。随后，我们将这些条件扩展到右端项也未知的情形；本文余下部分均采用这一设定。

命题*1*：假设存在 M 个总线性未知约束，其已知右端项为 d_i， i =**1**，.....， M，并且对于每个约束 $i = 1$，.....， M，存在 n_i 个满足 $T > n_i \ge n$ 的状态，使得该约束在时间步 $\mathcal{T}_i$处生效，其中 $\mathcal{T}_i = \{t | c_i^\top x_t = d_i\}$。如果对于每个约束， n 个这样的状态向量线性无关，那么约束向量 c_i 可以被精确恢复，对于 $i = 1,\dots,M$。

证明： 对于每个约束 i ，我们从约束处于绑定状态的 n_i 个状态中选取 n 个。于是有

$$C_{\mathcal{T}_i} \cdot X_{\mathcal{T}_i} = D_{\mathcal{T}_i}, \tag{6}$$

其中

$$C_{\mathcal{T}_i} = \left[I_n \otimes c_i^\top\right], X_{\mathcal{T}_i} = \begin{bmatrix} x_{t_{i,1}} \\ \vdots \\ x_{t_{i,n}} \end{bmatrix}, t_{i,1},\dots,t_{i,n} \in \mathcal{T}_i \text{ and}$$

$$D_{\mathcal{T}_i} = \left[\mathbf{1}_n d_i\right]. \tag{7}$$

对（6）重新排列，得到

$$\tilde{X}_{\mathcal{T}_i} \cdot c_i = D_{\mathcal{T}_i}, \tag{8}$$

其中

$$\tilde{X}_{\mathcal{T}_i} = \begin{bmatrix} x_{t_{i,1}}^\top \\ \vdots \\ x_{t_{i,n}}^\top \end{bmatrix}. \tag{9}$$

˜由于在$x_{t_{i,j}}, j = 1,\dots,n$线性无关的条件下， $X_{\mathcal{T}_i}$ 满行秩，因此可以精确恢复 c_i。非绑定状态可用于确定约束不等式的符号。 ■

在本文剩余部分，我们也将约束的右端项视为未知。为处理未知的右端项参数 d_i, $i \in \{1,\dots,M\}$，我们将状态维度扩展一个元素，该元素对应于约束的右端项，以简化记号。更具体地说， $x_t := [x_t, 1]^\top \in \mathbb{R}^{n+1}$ 和

$$A := \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}, B := \begin{bmatrix} B \\ 0 \end{bmatrix}. \tag{10}$$

现在，每个约束都可以简洁地写为 $c_i^\top x_t \le 0$where $c_i^\top := [c_i^\top, -d_i] \in \mathbb{R}^{n+1}$。在这种情况下，可以采用与命题 1 类似的方式，并在相同假设下进行精确约束推断。具体而言，可以将约束右端设为任意值，推断出的约束将只确定到一个比例因子。上述结果表明，只有在我们知道约束总数以及约束对哪些状态生效的强假设下，才能推断出精确约束。在本研究中，我们主要关注的是在更具挑战性的场景中提供一个约束推断框架：在这些场景中，实际约束数量以及约束生效的时间步（如果存在）都是未知的。为解决这一难题，我们采用以下 KKT 残差目标函数形式；这是一个通用框架，能够在最一般的设置下进行约束推断。

B. 通过 *KKT* 残差进行一般约束推断

本节关注实际约束数量 M 已先验确定的情况。我们将在第四节-D中将方法扩展到 M 未知的情形。接下来，我们针对来自一个初始条件的一条示范构建拉格朗日函数。只需将不同初始条件下多条示范对应的拉格朗日函数相加，即可轻松将这一设置扩展到多条示范。式（4）中的优化问题具有如下拉格朗日函数

$$\begin{aligned} L(U,\lambda) &:= (GU+Hx_0)^\top (I_T\otimes Q)(GU+Hx_0) \\ &+ U^\top (I_T\otimes R)U + \lambda^\top C(GU+Hx_0), \end{aligned} \tag{11}$$

其中 $\lambda \in \mathbb{R}_+^{MT}$表示拉格朗日乘子向量。KKT 条件中的驻点性、互补松弛性、原始可行性和对偶可行性随后

$$\nabla_U L(U^*,\lambda) = 2G^\top (I_T\otimes Q)(GU^*+Hx_0)+$$
$$2(I_T\otimes R)U^* + G^\top C^\top \lambda = 0 \tag{12}$$
$$\lambda^\top C(GU^*+Hx_0) = 0 \tag{13}$$
$$C(GU^*+Hx_0) \le 0 \tag{14}$$
$$\lambda \ge 0. \tag{15}$$

KKT残差公式通过最小化平稳性条件和互补松弛条件的某种度量，同时满足原始和对偶可行性，来获得未知参数的估计值；在我们的情形中，这些参数为 c_i、 $i=1$、.....、 M和 λ。更具体地说，

$$\begin{aligned} \min_{\{c_i\}_{i=1}^{i=M},\lambda} \quad & \ell(\{c_i\}_{i=1}^{i=M},\lambda) \\ \text{s.t.} \quad & C_{\{c_i\}_{i=1}^{M}} A_2 \le 0 \\ & \lambda \ge 0, \end{aligned} \tag{16}$$

为简洁起见，我们定义了其中

$$\ell(\{c_i\}_{i=1}^{i=M},\lambda) = \|A_1 + G^\top C_{\{c_i\}_{i=1}^{M}}^\top \lambda\|_2^2$$
$$+ \|(C_{\{c_i\}_{i=1}^{M}} A_2)^\top \lambda\|_2^2 + \rho_1 \sum_{i=1}^{M} \|c_i\|_2^2 + \rho_2 \|\lambda\|_2^2 \tag{17}$$
$$A_1 = 2G^\top (I_T\otimes Q)(GU^*+Hx_0) + 2(I_T\otimes R)U^* \tag{18}$$
$$A_2 = GU^* + Hx_0, \tag{19}$$

其中 ρ_1， ρ_2 为正则化参数。鉴于约束 c_i 现在未知，我们使用记号 $C_{\{c_i\}_{i=1}^{M}}$表示将 c_is映射如下的函数

$$C_{\{c_i\}_{i=1}^{M}} := \begin{bmatrix} I_T\otimes c_1^\top \\ \vdots \\ I_T\otimes c_M^\top \end{bmatrix}. \tag{20}$$

需要指出的是，这是一个关于 $c_i, i=1,\dots,M$ 和 λ [23] 的双凸优化问题。求解双凸优化问题最常用的方法是采用交替优化方案，我们将在下一节中概述该方案。

C. 交替优化公式

观察问题 (16) 可知，固定其中一个变量后，该问题便成为关于另一个变量的约束最小二乘问题。不失一般性，本节重点讨论以下情况： $M=1$ ，且唯一的约束为 $c \in \mathbb{R}^{n+1}$。更具体地说，固定 c 后，问题 (16) 变为

$$\begin{aligned} \min_{\lambda} \quad & \|\tilde{A}\lambda - \tilde{b}\|_2^2 + \rho_2\|\lambda\|_2^2 \\ \text{s.t.} \quad & \lambda \ge 0, \end{aligned} \tag{21}$$

其中

$$\tilde{A} = \begin{bmatrix} G^\top C_{\{c\}}^\top \\ (C_{\{c\}}A_2)^\top \end{bmatrix}, \tilde{b} = \begin{bmatrix} -A_1 \\ 0 \end{bmatrix} \text{ and } C_{\{c\}} = [I_T\otimes c].$$

另一方面，在对 c 进行最小化时，问题可以改写为 (16)

$$\begin{aligned} \min_{c} \quad & \|\tilde{\tilde{A}}c - \tilde{\tilde{b}}\|_2^2 + \rho_1||c||_2^2 \\ \text{s.t.} \quad & C_{\{c\}}A_2 \le 0, \end{aligned} \tag{22}$$

其中

$$\tilde{\tilde{A}} = \begin{bmatrix} G^\top \Lambda \\ A_2^\top \Lambda \end{bmatrix}, \tilde{\tilde{b}} = \begin{bmatrix} -A_1 \\ 0 \end{bmatrix},$$

且 Λ 是一个构造的稀疏矩阵，其元素为 λ 的坐标，使得 $C_{\{c\}}^\top \lambda = \Lambda c$。使用矩阵 Λ 进行重新表述，完全是为了展示问题 22 确实是关于 c 的最小二乘问题，其约束对 c 也呈线性。

问题 (21) 和 (22) 的目标函数中包含正则化项，以确保解的唯一性。交替优化方法的详细流程见算法 1。我们交替求解问题 (21) 和 (22)，迭代 K 次。

算法 1 交替最小化算法 thm

1: **随机初始化：** $c \in \mathbb{R}^{n+1}$
2: **对于** $K=1,\cdots$ **执行**
3: 求解问题 (21)
4: 更新 λ 值
5: 求解问题（22）
6: 更新 c值
7: **结束循环**
8: **输出：** c, λ

定理 *1*: 算法 1 收敛到 (16) 的一个驻点。

证明: 要使该定理成立，问题 (21) 和问题(22)必须具有唯一解 [24]。问题 (21) 是一个正则化约束最小二乘问题，其目标函数是严格凸的。因此，问题 (21) 存在唯一解。类似地，由于强凸性，问题(22)也存在唯一解。随后根据 [24] 中的命题 2.7.1可得收敛性。该结果针对仅推断一个约束的情况给出，因为正如将在后续章节中明确的那样，我们利用算法 1 一次推断一个约束。

■

D. 贪心约束推断

前几节讨论了环境中约束数量先验已知但约束未知的情形。然而，在现实场景中，这类情况很少见，因此我们需要一种启发式方法，在约束数量未知时对约束进行推断。为此，我们提出使用 KKT残差目标函数作为约束推断质量的估计。在下文中，我们提出一种贪心约束推断算法，逐步添加约束，直至满足某一判据。

增量式贪心约束推断（IGCI）方法如算法 2 所示，每次迭代都利用算法 1 在推断约束集合中逐步添加一个约束。该方法从一个空约束集合 $\mathcal{C}$ 开始。按照算法 1 推断出一个约束 c_i 和拉格朗日乘子后，会将该约束暂时添加到约束集合 $\mathcal{C}$ 中。随后评估 KKT残差目标函数 $\ell(\{c_i\}_{c_i\in\mathcal{C}}, \lambda)$。如果该值与上一轮迭代的 KKT 残差（在算法 2 中记为 E ）之间的差值高于用户指定的阈值 δ，则约束 c_i 将保留在约束集合中。重复此过程，直至该差值低于预设阈值，此时过程终止，且不将最后推断出的约束纳入 $\mathcal{C}$。算法终止时，

算法 2IGCI

1: **参数：** 收敛阈值 δ, $E \gg 1$
2: **初始化：** $\mathcal{C}' = \{\}$
3: **对于** $N_c = 1, \ldots$ **执行**
4: 获取 $c_{N_c} \in \mathbb{R}^{n+1}$ 和 $\lambda \in \mathbb{R}^{N_c T}$ 来自算法 1
5: $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{c_{N_c}\}$
6: $\lambda' \leftarrow \lambda$
7: **如果** $|\ell(\{c\}_{c\in\mathcal{C}'}, \lambda') - E| \leq \delta$ **那么**
8: 中断
9: **else**
10: $\mathcal{C} \leftarrow \mathcal{C}'$, $\lambda \leftarrow \lambda'$
11: $E \leftarrow \ell(\{c\}_{c\in\mathcal{C}}, \lambda)$
12: **结束如果**
13: **结束循环**
14: **输出：** $\mathcal{C}, \lambda$

一组 N_c 推断约束 $\mathcal{C} = \{\tilde{c}_1, \ldots, \tilde{c}_{N_c}\}$ 将被返回。应注意，N_c 不必等于原始约束数量 M。此外，鉴于第 7行中的判据在第一次迭代时不会得到满足，IGCI将至少添加一个约束。如有需要，可以通过为 E适当选择一个更小的初始值来进行修改。最后，在第 4 行中，可以多次独立运行算法 1，以获得更好的解。

在 IGCI 的任意一次迭代 k 中，第 4 行会获得一个额外约束和拉格朗日乘子的估计值。具体而言，使用算法 1 求解式（16），其中未知量为 c_k 和乘子 $\lambda \in \mathbb{R}^{kT}$，同时固定已推断约束的值 $\tilde{c}_i$, $i =$ 1、.....、$k-1$。尽管在每次迭代中进一步固定已经推断出的拉格朗日乘子、仅推断一个额外的 $\lambda \in \mathbb{R}^T$ 也能取得良好效果，但我们的仿真表明，在第 k 次迭代中推断所有约束 $i = 1$、... ...、k对应的拉格朗日乘子，可以获得略高的准确性。

最后，到目前为止的分析都是基于单个最优示范 U^*。在后续章节中，我们通过对问题 (21) 和 (22) 中的 ~ ~ ~ ~ ~ ~A, b,A和 b作适当修改，允许根据多个示范进行推断。更具体地说，对于从状态 $x_0^{(1)}, \ldots, x_0^{(N)}$ 出发的 N个示范 $U_1^*, \ldots, U_N^*$，并记为

$$A_1^{(i)} = 2G^\top (I_T \otimes Q)(GU_i^* + Hx_0^{(i)}) + 2(I_T \otimes R)U_i^* \tag{23}$$

$$A_2^{(i)} = GU_i^* + Hx_0^{(i)}, \tag{24}$$

~ ~问题21的新 A 和 b现在变为

$$\tilde{A} = \begin{bmatrix} G^\top C_{\{c\}}^\top \\ (C_{\{c\}} A_2^{(1)})^\top \\ \vdots \\ G^\top C_{\{c\}}^\top \\ (C_{\{c\}} A_2^{(N)})^\top \end{bmatrix}, \tilde{b} = \begin{bmatrix} -A_1^{(1)} \\ 0 \\ \vdots \\ -A_1^{(N)} \\ 0 \end{bmatrix}.$$

.....的定义在线a ~ 与 ~ ~ ~ ~ ~

~ ~分别对应 $\mathcal{A}$ 和 $\mathcal{A}^c$。现在，我们将无约束集合的覆盖率定义为 $\frac{Vol(\mathcal{A}^c \cap \tilde{\mathcal{A}}^c)}{Vol(\mathcal{A}^c)}$，并将推断出的无约束集合与实际受约束集合的重叠度定义为 $\frac{Vol(\tilde{\mathcal{A}}^c \setminus \mathcal{A}^c)}{Vol(\mathcal{A})}$。这些指标可视为类似于真正例（TP）和假正例（FP）的分类率，因此将称其为真正例和假正例。

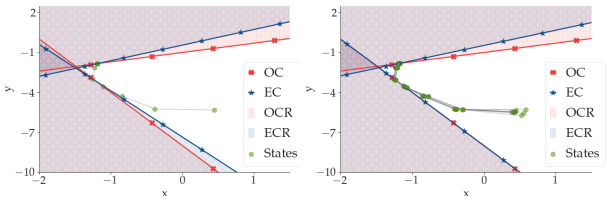
A. 二维导航任务

在这组仿真中，我们研究一个具有 $x_t \in \mathbb{R}^2, u_t \in \mathbb{R}$、动力学矩阵的二维线性系统。

$$A = \begin{bmatrix} 0.8 & 0.1 \\ 0.1 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix}, \tag{27}$$

以及二次代价矩阵 $Q = 100 \cdot I_2$ 和 $R = 0.1$。控制器的目标是在任务期间始终接近预定轨迹 x_{track}，该轨迹列于附录 A 中。我们在具有两个线性状态约束的环境中评估算法 1，如图 1 和图 2 所示。控制任务的时域为 $T = 8$，初始状态围绕 $[-0.5, -5.5]^\top$ 按$\sigma = 0.05$分布。我们考虑两种情况：一种具有完美状态观测，另一种具有不完美状态观测。在所有仿真中，我们改变专家示范的数量 N，以量化其对推断的影响。IGCI 在 2 次迭代后终止，因为对于后续迭代，KKT 代价函数（如图 7a 所示）不再显著下降。对于本节中的所有仿真，我们遵循肘部法则来终止算法，而不显式选择阈值 δ。对于每个实验，我们使用新获得的专家示范进行 10 次相互独立的仿真。

我们首先研究完美状态观测下的约束推断。在图 1 中，我们绘制了 $N = 1$ 和 $N = 10$ 的专家示范，以及实际约束区域和估计约束区域。我们分别用 OC 和 OCR 表示原始仿射约束及其对应区域。EC 和 ECR 表示估计仿射约束和约束区域。



(a) $N = 1$. (b) $N = 10$.

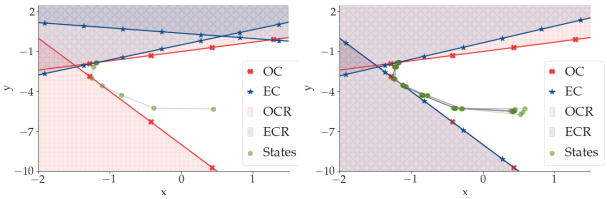
图1：在完美状态观测下，$N = 1$（图1a）和 $N = 10$（图 1b ）的原始和估计受约束区域，以及专家示范 under。

对于含噪状态观测的情况，我们假设每个演示 i 都受到如下观测噪声的干扰

$$\tau_i = \{x_0, u_0, x_1 + \epsilon, u_1, x_2 + \epsilon, \ldots, u_{T-1}, x_T + \epsilon\}, \tag{28}$$

其中 $\epsilon \sim \mathcal{N}(0, 0.005)$。在图 2 中，我们绘制了原始约束以及一组推断约束实例，适用于 $N = 1$ 和 $N = 10$，分别。对于 $N = 1$，我们展示了 10 个仿真中的一个，其中对某个的推断

约束推断不准确。显然，无论是完美的状态观测还是含噪状态观测，更多的示范（这可能会导致与约束进行更多交互）都会带来更准确的推断。图3展示了真阳性率和假阳性率，下一节将对此进行讨论。



(a) $N = 1$. (b) $N = 10$.

图 2： $N = 1$ （图 2a）和 $N = 10$ （图 2b）的原始与估计受约束区域示例，以及带噪声状态观测下的专家示范。

B. 基线比较

为强调推断约束参数而不仅仅是安全/不安全区域的益处，我们将 IGCI 与 [4]中提出的推断方法进行比较。后者推断状态空间中被认为是受约束或无约束的区域。我们求解 [4] 中的问题5，以便在完美和不完美状态观测的情况下，为我们的二维导航任务获得无约束区域的估计。我们利用10 个不同的查询点来获得无约束集合的估计。算法的评估使用真阳性和

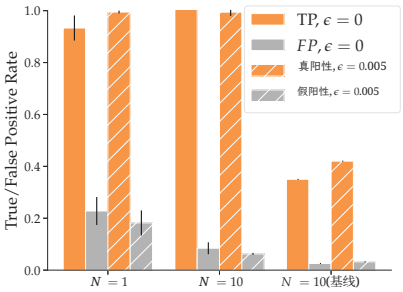
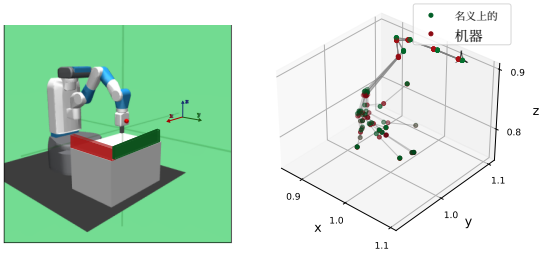


图 3：IGCI和基线在完美状态观测和含噪状态观测下的分类率。结果取10 次仿真的平均值。当 $N = 10$时，我们的方法具有显著高于基线方法的真正例率，同时保持同等水平的假阳性率。

假阳性分类率。

无噪声和有噪声状态观测的结果均见图3。增加可用轨迹的数量自然会带来更准确的估计，从而能够制定更安全的策略。基线方法只在较有利的 $N = 10$ 示范情形下实现，它只能识别轨迹周围的矩形区域，因此未能成功。

以识别示范“右侧”的大多数无约束区域。另一方面，我们的方法能够推断出大部分无约束区域，同时其假阳性率与更为保守的基线相近。需要强调的是，我们的方法返回约束的完整参数化形式，因此，新的前向优化问题仍然是凸的。另一方面，基线方法推断出矩形区域，而这些区域的并集不一定构成凸集，因此，在推断得到的定义域中处理新的前向问题时需要采用更复杂的技术。最后，基线方法需要求解非线性规划或混合整数线性规划，而与我们的交替方法相比，这可能会显著降低效率。



(a) Fetch-Reach机器人。 (b) 末端执行器轨迹。

图 4：(4a) Fetch-Reach机器人及其约束（红色、绿色）。(4b) $N = 10$ 的标称轨迹及对应的实际机器人轨迹。

C. 采用次优示范的三维操作任务

在本节中，我们使用 Fetch-Reach 机器人模拟器 [25]，该模拟器旨在三维环境中完成推动和抓取物体等任务。控制输入是机器人夹爪在 x, y 和 z 坐标中的期望位移，$x_t \in \mathbb{R}^3, u_t \in \mathbb{R}^3$。我们假设名义动力学矩阵（即机器人末端执行器理论上遵循的动力学矩阵）为 $A = I_3$ 和 $B = I_3$，二次代价矩阵为 $Q = 10 \cdot I_3$ 和 $R = 10 \cdot I_3$。

需要注意的是，名义动力学是对实际动力学的简化，因为底层物理引擎要复杂得多；对于特定的控制输入，机器人夹爪不会落在名义模型预测的状态，而是落在与该状态相近的状态上。因此，我们研究两种情况：一种使用名义轨迹，另一种使用实际机器人轨迹。为了获得后者，在利用名义动力学求解（3）得到最优轨迹后，我们在各个最优航点之间使用闭环控制器，引导夹爪接近名义状态（机器人状态）。对于每个航点，控制器运行 5 次，其动作由当前状态与航点之间的差异决定。显然，这些如图4b所示的轨迹，根据（3）已不再是最优轨迹。

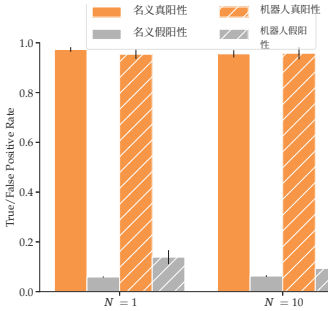


图 5：标称轨迹和机器人轨迹的分类率。由于Fetch-Reach仿真环境中标称动力学与真实机器人动力学之间存在模型差异，机器人轨迹被视为次优轨迹 [25]。结果取 10 次独立仿真的平均值。即使轨迹是次优的，我们的方法仍然有效。

该控制任务要求末端执行器在时域 $T = 8$ 内尽可能贴近跟踪轨迹，该轨迹见附录 A。图 4a 示意展示的真实值潜在约束为 $x \geq 0.9$ 和 $y \leq 1.1$，也可以写作

$c_i^\top x \leq 0, \quad i = 1, 2$，其中有 $c_1 = [-1, 0, 0, 0.9]^\top$ 和 $c_2 = [0, 1, 0, -1.1]^\top$。整个定义域的边界为 $x \in [0.4, 1.4]^\top$，$y \in [0.4, 1.4]^\top$和 $z \in [0.4, 1.4]^\top$。我们收集了 $N = 1$ 和 $N = 10$专家轨迹，其初始状态以 $[1, 1, 0.8]$ 为中心呈正态分布，$\sigma = 0.05$。图4b展示了$N = 10$ 轨迹，其中绿色点对应名义轨迹，红色点则表示实际机器人状态。我们同时使用名义示范和次优示范来推断约束，以便从次优性的角度量化我们方法的性能。我们运行了10次独立仿真，并在图5中报告平均分类率。显然，估计

即使只利用单个专家示范也能取得良好效果。图 6 绘出了针对 $N = 10$ 条名义轨迹得到的实际约束和推断约束。图 7b 展示了 KKT 残差目标函数随迭代次数变化的收敛率。正如预期的那样，对于最优示范，KKT 代价函数低于次优示范对应的值。

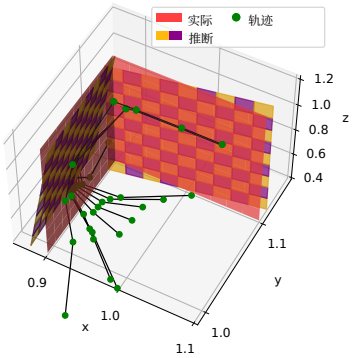
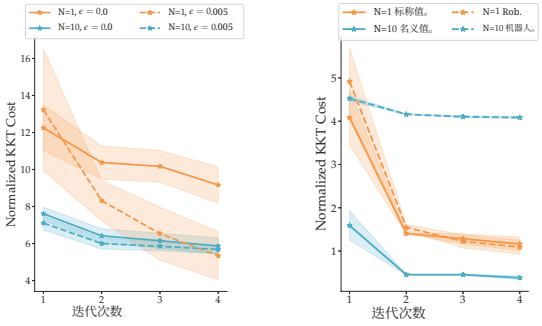


图 6：末端执行器的标称轨迹（$N = 10$），以及实际约束和推断约束。推断约束与实际约束吻合良好。



(a) (b)

图 7：二维导航任务（图 7a）和三维操作任务（图 7b）中，KKT残差代价函数随迭代次数的变化，按轨迹数归一化。结果为 10 次仿真的平均值。

VI. 结论与未来拓展

在这项工作中，我们提出了一种方法，通过观测专家示范来推断控制任务中的约束。我们展示了能够实现精确恢复的条件。我们提出通过最小化 KKT残差目标函数来学习未知约束。我们还介绍了 IGCI，这是一种算法，可用于在除约束的参数化表示之外没有其他信息的情况下推断约束。我们通过多项仿真评估了 IGCI 的性能，并展示了在完美、有噪声和次优观测下，推断出的约束能够保持准确而不危及安全。

本文提出的交替最小化过程足够灵活，可以处理能够建模为半空间的并集与交集的更复杂约束。另一个有前景的方向是在潜在的非线性控制问题中，设计控制输入，以主动收集关于未知凸约束甚至非凸约束的信息。

参考文献

[1] A. Y. Ng、S. Russell 等，“逆强化学习算法”。载于 *Icml*，第1卷，2000年，第2页。[2] A. Hussein、M. M. Gaber、E. Elyan 和 C. Jayne，“模仿学习：学习方法综述”，*ACM Computing Surveys (CSUR)*，第50卷第2期，第1–35页，2017年。[3] G. Liu、Y. Luo、A. Gaurav、K. Rezaee 和 P. Poupart，“逆强化学习中的约束推断基准测试”，*arXiv preprint arXiv:2206.09670*，2022年。[4] G. Chou、N. Ozay 和 D. Berenson，“在成本函数不确定性下从局部最优示范中学习约束”，*IEEE Robotics and Automation Letters*，第5卷第2期，第3682–3690页，2020年。[5] T. C. Chan、R. Mahmood 和 I. Y. Zhu，“逆优化：理论与应用”，*arXiv preprint arXiv:2109.03920*，2021年。[6] D. R. Scobee 和 S. S. Sastry，“逆强化学习中的最大似然约束推断”，*arXiv preprint arXiv:1909.05477*，2019年。[7] S. Malik、U. Anwar、A. Aghasi 和 A. Ahmed，“约束逆强化学习”，载于 *International Conference on Machine Learning*。PMLR，2021年，第7390–7399页。[8] D. Papadimitriou、U. Anwar 和 D. S. Brown，“强化学习中的约束推断贝叶斯方法”，*Transactions on MachineLearning Research*。

[9] C. P´erez-D’Arpino 和 J. A. Shah，“C-learn：从示范中学习几何约束，用于共享自主中的多步操作”，载于 *2017*年*IEEE*国际机器人与自动化会议（*ICRA*）。IEEE，2017年，第4058–4065页。[10] G. Chou、D. Berenson 和 N. Ozay，“从示范中学习约束”，载于 *International Workshop on the Algorithmic Foun- dations of Robotics*。Springer，2018年，第228–245页。[11] A. Robey、H. Hu、L. Lindemann、H. Zhang、D. V. Dimarogonas、S. Tu 和 N. Matni，“从专家示范中学习控制屏障函数”，载于*2020*年第*59*届*IEEE*决策与控制会议（*CDC*）。IEEE，2020年，第3717–3724页。[12] T. L. Molloy、J. J. Ford 和 T. Perez，“有限与无限时域上控制受限离散时间系统的在线逆最优控制”，*Automatica*，第120卷，第109109页，2020年。[13] A. Agrawal、S. Barratt 和 S. Boyd，“学习凸优化模型”，*IEEE/CAA*《自动化学报》，第8卷第8期，第1355–1364页，2021年。[14] M. Menner、P. Worsnop 和 M. N. Zeilinger，“约束逆最优控制及其在人类操作任务中的应用”，*IEEE*《控制系统技术汇刊》，第29卷第2期，第826–834页，2019年。[15] J. Li、C.-Y. Chiu、L. Peters、S. Sojoudi、C. Tomlin 和 D. Fridovich-Keil，“从含噪的部分状态观测和不完整轨迹中推断反馈动态博弈的成本”，*arXiv preprint arXiv:2301.01398*，2023年。[16] R. K. Ahuja 和 J. B. Orlin，“逆优化”，《运筹学》，第49卷第5期，第771–783页，2001年。[17] T. C. Chan 和 N. Kaw，“用于恢复约束参数的逆优化”，《欧洲运筹学杂志》，第282卷第2期，第415–427页，2020年。[18] K. Ghobadi 和 H. Mahmoudzadeh，“利用逆优化推断线性可行域”，《欧洲运筹学杂志》，第290卷第3期，第829–843页，2021年。[19] P. Englert、N. A. Vien 和 M. Toussaint，“逆KKT：从示范中学习操作任务的成本函数”，*The Interna- tional Journal of Robotics Research*，第36卷第13–14期，第1474–1488页，2017年。[20] C. Awasthi，“最优控制与动态博弈论中的正向与逆向方法”，博士学位论文，明尼苏达大学，2019年。[21] M. Menner 和 M. N. Zeilinger，“最优控制器逆向学习的最大似然方法”，*IFAC-PapersOnLine*，第53卷第2期，第5266–5272页，2020年。[22] A. Keshavarz、Y. Wang 和 S. Boyd，“填补凸目标函数”，载于 *2011 IEEE international symposium on intelligent con- trol*。IEEE，2011年，第613–619页。[23] J. Gorski、F. Pfeuffer 和 K. Klamroth，“双凸集与双凸函数优化：综述与扩展”，*Mathematical methods of operations research*，第66卷第3期，第373–407页，2007年。[24] D. P. Bertsekas，非线性规划。Athena Scientific，1999年。[25] R. de Lazcano、K. Andreas、J. J. Tai、S. R. Lee 和 J. Terry，“Gymnasium Robotics”，2023年。[在线]。可访问：http://github.com/ Farama-Foundation/Gymnasium-Robotics

附录

A. 仿真详情

尽管IGCI 对ρ_1、ρ_2的选择并不十分敏感，但在所有仿真中，我们都使用 $\rho_1 = 10$ 和 $\rho_2 = 0$。二维导航任务：跟踪轨迹为 x_{track} =[0., −5.5； −0.6、 −4.5； −1.5、 −3.8； −1.5、 −3.5； −1.6、 −2.5； − 0.2、 −2； 0.5、 −2； 0.8、 −1.5]$^\top$。Gym环境：跟踪轨迹为 x_{track} =[0.95, 1, 0.8； 0.9,1,0.8； 0.85,1,0.8； 0.8,1.1,0.9； 0.9,1.15,0.9； 0.95,1.2, 0.9； 1, 1.25, 0.9； 1.1, 1.3, 0.9]$^\top$。