Notes on Sample Complexity of IRL on Constrained MDPs and Bandit problems setting

Titouan Renard

May 1, 2023

Contents

1 Constrained and Unconstrained MDPs **2**
1.1 Unconstrained Markov Decision Processes . . . 2
1.1.1 Discounted Reward, Value, Quality and Objective Functions . 3
1.1.2 The Occupancy Measure . . . 4
1.1.3 Formulating solving MDPs as optimization problems . . . 6
1.2 Constrained Markov Decision Processes . . . 8
1.2.1 Discounted Costs, Cost-Value, Cost-Quality, Cost-Objective function . . . 8
1.2.2 Feasibility . . . 9
1.2.3 Goal of the CMDP, formulating optimization problems . . . 9
1.2.4 Solution Maps . . . 9
1.3 Inverse Reinforcement Learning . . . 11

2 Min-Max Optimization and Saddle Point Problems **13**
2.1 Preliminaries . . . 13
2.2 Gradient Descent Ascent . . . 15
2.3 Extra-Gradient Descent Ascent . . . 17
2.4 Optimistic Gradient Descent Ascent . . . 21
2.5 Summary of Known Results . . . 21

3 Properties of regularizers **22**
3.1 Shannon Entropy . . . 22
3.1.1 Shannon Shenanigans: Lipschitzness of the Shannon Entropy over $\Delta_n^{(\rho)}$ . . . 23
3.2 Rényi Entropy . . . 25

4 Constrained and Unconstrained Bandits, Iteration Complexity for different Regularizers **27**
4.1 A few observations about Bandit problems . . . 27
4.2 Constrained Inverse bandit (CIB) . . . 28
4.2.1 Solving the Shannon Entropy regularized problem with an extra-gradient approach: EG-COP . . . 28
4.2.2 Solving the Shannon-regularized problem with gradient descent-ascent : GDA-COP . . . 32

1 Constrained and Unconstrained MDPs

1.1 Unconstrained Markov Decision Processes

Definition 1.1. *(**Markov Decision Process**) We define a **Markov Decision Process** (MDP) as the following 6-tuple:*

$$M = (S, A, P, \boldsymbol{\nu}, \boldsymbol{r}, \gamma),$$

*where $S = \{s_1, s_2, ..., s_n\}$ is a discrete set of **states** of size $|S| = n$, $A = \{a_1, a_2, ..., a_m\}$ a discrete set of **actions** of size $|A| = m$, is a probabilistic markovian transition law $P : S \times A \to \Delta_S$ that we can represent as a $\mathbb{R}^{nm \times n}$ matrix (where each column represents in the distribution $\in \Delta_S$), $\nu \in \Delta_s$ is the **initial distribution** of states at the start of the process, $r : S \times A \to \mathbb{R}$ is a reward function that we can conveniently represent as a vector $\boldsymbol{r} \in \mathcal{R} \subseteq \mathbb{R}^{nm}$ since S and A are discrete (we call $\mathcal{R}$ the **reward class**), finally $\gamma \in (0, 1)$ is our **discount factor**.*

Unless specified otherwise it is expected that Markov Decision Process run for an infinite time, each (infinite-time) sequence of move on the MDP is called a *trajectory* and is denoted:

$$\tau = \{s_1, a_1, s_2, a_2, ...\},$$

where the transition from one state to another is governed by the markovian transition law:

$$s_{t+1} \sim P(\cdot|s_t, a_t).$$

The MDP, doesn't specify how actions are selected, in practice, this is done by choosing a *policy function* $\pi : S \to \Delta_A$, which we use to pick the action a_t:

$$a_t \sim \pi(s_t) \in \Delta_A.$$

Definition 1.2. *(**Policy**) A **policy** is a function $\pi : S \to \Delta_A$ that maps each discrete state to a distribution over actions of an MDP M. Because we consider discrete state and action sets, we can represent policies as matrices $\pi \in \Pi \in \mathbb{R}^{n \times m}$. Where Π denotes the set of **valid policies**, i.e. :*

$$\Pi := \Big\{\pi = [\pi_{s_1}, \pi_{s_2}, ..., \pi_{s_n}], \pi_{s_i} \in \Delta_A \forall s_i \in S\Big\}.$$

Assuming actions are picked from some fixed policy function $\boldsymbol{\pi}\Pi$ the progression from state to state of the MDP becomes a Markov Chain described by the closed-loop transition law.

Definition 1.3. *(**Closed-Loop Transition Law**) We define the **closed-loop transition law** $P^\pi : S \to \Delta_S$ as a function that gives the distribution over the next state as a function of the current state, this distribution is trivially given by:*

$$P^\pi(s'|s) = \sum_a P(s'|a, s) \cdot \pi(a|s).$$

It can be represented as a square matrix $P^\pi \in \mathbb{R}^{n \times n}$, where each i-th column gives the distribution of states s' associated with the transition from the i-th state.

1.1.1 Discounted Reward, Value, Quality and Objective Functions

At each transition the agent receives reward $r_t = r(s_t, a_t)$. We want to quantify the general performance of our agents with respect to that reward function, across complete trajectories. This motivates the definition a series of functions which are used to quantify the performance of a given policy π on the MDP.

Definition 1.4. *(**Discounted Reward**) Given a policy π, and a convex **policy regularizer function** $\Omega : \Pi \to \mathbb{R}$, one can compute the discounted reward $R(\tau)$ as follows:*

$$R(\tau) = (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} \gamma^t \Big(r(s_t, a_t) - \Omega(\pi(\cdot|s_t)) \Big); \ s_t, a_t \in \tau.$$

Note that assuming that the regularized rewards $(r(s_t, a_t) - \Omega(\pi(\cdot|s_t)))$ are upper-bounded by some constant $C \in \mathbb{R}$, this limit is guaranteed to exist as:

$$R(\tau) = (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} \gamma^t \Big(r(s_t, a_t) - \Omega(\pi(\cdot|s_t)) \Big) \leq (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} \gamma^t C = C.$$

Since the process is stochastic it makes more sense to think in terms of *expected discounted reward*, we define the *objective function* $J(\boldsymbol{\pi}, \boldsymbol{r})$, the *value function* $V_r^\pi(s)$ and the *quality function* $Q_r^\pi(s, a)$ (which are all associated with a policy π and a reward function r).

Definition 1.5. *(**Value Function**) For a given policy $\pi : S \to \Delta A$, a convex **policy regularizer function** $\Omega : \Pi \to \mathbb{R}$ and under the assumption that the agent starts at some state s, we define the **value function** $V_r^\pi : S \to \mathbb{R}$ as:*

$$\begin{aligned} V_r^\pi(s) &= \mathbb{E}\Big[R(\tau) \Big| s_0 = s, \pi \Big] \\ &= \mathbb{E}\left[(1-\gamma) \sum_{t=0}^{+\infty} \gamma^t \Big(r(s_t, a_t) - \Omega(\pi(\cdot|s_t)) \Big) \Big| s_0 = s, \pi \right]. \end{aligned}$$

Definition 1.6. *(**Quality Function**) For a given policy $\pi : S \to \Delta A$, a convex **policy regularizer function** $\Omega : \Pi \to \mathbb{R}$ and under the assumption that the agent starts at some state s with some first action a, we define the **quality function** $Q_r^\pi : S \times A \to \mathbb{R}$ as:*

$$\begin{aligned} Q_r^\pi(s, a) &= \mathbb{E}\Big[R(\tau) \Big| s_0 = s, a_0 = a, \pi \Big] \\ &= \mathbb{E}\left[(1-\gamma) \sum_{t=0}^{+\infty} \gamma^t \Big(r(s_t, a_t) - \Omega(\pi(\cdot|s_t)) \Big) \Big| s_0 = s, a_0 = a, \pi \right]. \end{aligned}$$

Definition 1.7. *(**Objective Function**) For a given policy $\pi : S \to \Delta A$, convex policy regularizer function $\Omega : \Pi \to \mathbb{R}$ and and under an initial state distribution $\boldsymbol{\nu} \in \Delta_S$ we define the **objective function** as:*

$$J(\pi, r) = \mathbb{E}\Big[R(\tau) \Big| s_0 \sim \nu, \pi \Big] = \mathbb{E}\Big[V^\pi(s) \Big| s \sim \nu \Big].$$

*Since we most often use the **objective function** for optimization in policy space, we consider that $J : \Pi \times \mathcal{R} \to \mathbb{R}$ is a function of the policy π and the reward r.*

1.1.2 The Occupancy Measure

We now discuss a descriptor of the distribution of an agent's position across all states induced by some policy π: the *occupancy measure* is defined as follows.

Definition 1.8. *(**Occupancy Measure**) The (state-action) **occupancy measure** $\boldsymbol{\mu}^\pi \in \Delta_{S\times A}$ is defined for a given state-action pair as follows:*

$$\begin{aligned}\mu^\pi(s,a) &= (1-\gamma)\lim_{T\to+\infty}\sum_{t=0}^{T}\gamma^t\mathbb{P}^\pi_\nu(s_t=s,a_t=a)\\ &= (1-\gamma)\mathbb{E}\Big[\sum_{t=0}^{+\infty}\gamma^t\mathbf{1}(s_t=s,a_t=a)\Big|s_0\sim\nu,\pi\Big],\end{aligned}$$

since our state and action spaces are discrete we can represent $\boldsymbol{\mu}^\pi$ as a vector in $\mathbb{R}^{nm}$.

Definition 1.9. *(**Occupancy Set**) The **set of valid occupancy measure** (or occupancy set) is given by:*

$$\mathcal{M} := \Big\{\mu\in\mathbb{R}^{nm}_+ : (E-\gamma P)^T\boldsymbol{\mu} = (1-\gamma)\boldsymbol{\nu}\Big\}.$$

*The equality that all points $\boldsymbol{\mu}$ in the set must satisfy are known as **Bellman Flow Constraints**.*

Definition 1.10. *(**State-Occupancy Measure**) Similarly to the (state-action) occupancy measure, one can define a **state-occupancy measure** $\boldsymbol{\mu}^\pi_S \in \Delta_S$ is defined for a given state as follows:*

$$\begin{aligned}\mu^\pi_S(s) &= (1-\gamma)\lim_{T\to+\infty}\sum_{t=0}^{T}\gamma^t\mathbb{P}^\pi_\nu(s_t=s)\\ &= (1-\gamma)\mathbb{E}\Big[\sum_{t=0}^{+\infty}\gamma^t\mathbf{1}(s_t=s)\Big|s_0\sim\nu,\pi\Big],\end{aligned}$$

since our state and action spaces are discrete we can represent $\boldsymbol{\mu}_S$ as a vector in $\mathbb{R}^n$.

Furthermore, when an state action pair s,a has occupancy measure $\mu^\pi(s,a)=0$ (under policy π), we say that it is *unvisited*.

Observation 1.1. *The occupancy measure provides us with a very concise notation for writing out quantities such as the **objective function**, which can be represented in a simple scalar product form:*

$$J(\boldsymbol{\pi},\boldsymbol{r}) = \langle\boldsymbol{r},\boldsymbol{\mu}^\pi\rangle - \tilde{\Omega}(\boldsymbol{\mu}^\pi).$$

Which can be easily verified as follows:

$$
\begin{aligned}
J^{\nu}(\pi) &= (1-\gamma)\mathbb{E}\Big[R(\tau)\Big|s_0 \sim \nu, \pi\Big] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t(r(s,a) - \Omega(\pi(\cdot|s)))\Big|s_0 \sim \nu, \pi\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{s,a}\sum_{t=0}^{+\infty}\gamma^t r(s,a)\,\mathbf{1}(s,a)\Big|s_0 \sim \nu, \pi\right] \\
&- (1-\gamma)\mathbb{E}\left[\sum_{s}\sum_{t=0}^{+\infty}\gamma^t \Omega(\pi(\cdot|s)))\,\mathbf{1}(s)\Big|s_0 \sim \nu, \pi\right] \\
&= \sum_{s,a} r(s,a)\overbrace{(1-\gamma)\mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t\,\mathbf{1}(s,a)\Big|s_0 \sim \nu, \pi\right]}^{=\boldsymbol{\mu}^{\pi}(s,a)} \\
&- \sum_{s}\Omega(\pi(\cdot|s))\overbrace{(1-\gamma)\mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t\,\mathbf{1}(s)\Big|s_0 \sim \nu, \pi\right]}^{=\boldsymbol{\mu}_S^{\pi}(s)} \\
&= \sum_{s,a} r(s,a)\boldsymbol{\mu}^{\pi}(s,a) - \overbrace{\sum_{s}\Omega(\pi(\cdot|s))\boldsymbol{\mu}_S^{\pi}(s)}^{:=\tilde{\Omega}(\boldsymbol{\mu})} \\
&= \boldsymbol{r}^T\boldsymbol{\mu}^{\boldsymbol{\pi}} - \tilde{\Omega}(\boldsymbol{\mu}^{\pi}).
\end{aligned}
$$

Lemma 1.1. *(**Expected-Regularizer is convex in** μ) Consider the function* $\tilde{\Omega} := \mathbb{E}_{s\sim\boldsymbol{\mu}}[\Omega(\pi(\cdot|s))]$,

- *if* Ω *is* ***convex****, then so is* $\tilde{\Omega}$
- *if* Ω *is* ***strictly-convex****, then so is* $\tilde{\Omega}$.

(Proof in [6].)

Observation 1.1, together with lemma 1.1 is quite important as it shows that the objective function is **concave** w.r.t the occupancy measure (which opens the door for optimization approaches), moreover in the special case where $\Omega(\pi(\cdot|s)) = 0,\ \forall s \in S$ the objective is **linear** w.r.t the occupancy measure (which opens the door for linear programming approaches).
For convenience we interchangeably write the objective as a function of the policy π and of the occupancy measure $\boldsymbol{\mu}$:

$$J^{\nu}(\pi) = J^{\nu}(\boldsymbol{\mu}).$$

Observation 1.2. *There exists a (almost) well-defined bijection between any policy* π *and it's associated occupancy measure* $\boldsymbol{\mu}^{\pi}$*. With knowledge of* P *and* π*, one can compute the occupancy measure from the bellman flow constraint as follows:*

1. *compute the* ***closed loop transition law*** P^π,

2. *compute the* ***state-occupancy measure*** *from the occupancy measure, this is easy as it has a matrix-geometric series form:*

$$\mu_S^\pi = (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} (\gamma P^\pi)^t = (1-\gamma)(I - \gamma P^\pi)^{-1}$$

3. *finally one can compute the* ***occupancy measure*** *by*

$$\mu(s,a) = \pi(a|s)\mu_S(s).$$

Similarly, with knowledge of the occupancy mapping $\boldsymbol{\mu}$ *one can recover a policy* π *as follows:*

$$\pi^{\boldsymbol{\mu}}(a|s) = \begin{cases} \frac{\mu(s,a)}{\sum_a \mu(s,a)}, & \sum_a \mu(s,a) > 0 \\ \frac{1}{m}, & \textit{otherwise.} \end{cases}$$

Notice that the mapping is ill-defined for unvisited states, but this does not matter changes in policy on unvisited states have no impact on the discounted reward.

For convenience we define two maps that allow us to go from occupancy map to policies and from policies to occupancy maps.

Definition 1.11. *The* $OM : \Pi \to \mathcal{M}$ *returns the occupancy measure* $\boldsymbol{\mu} = OM(\boldsymbol{\pi})$ *associated with a given policy* $\boldsymbol{\pi}$.

Definition 1.12. *The* $PO : \mathcal{M} \to \Pi$ *returns a policy* $\boldsymbol{\pi} = PO(\boldsymbol{\mu})$ *that induces the occupancy measure* $\boldsymbol{\mu}$.

We expect that the policy for unvisited state is always uniform ($\pi(a|s_{\text{univisted}}) = 1/m \ \forall a \in A$), the following property is satisfied

$$(\text{OM} \circ \text{PO}) = \text{Id}$$
$$(\text{PO} \circ \text{OM}) = \text{Id}$$

where Id is the identity operator.

1.1.3 Formulating solving MDPs as optimization problems

The goal of an MDP can be stated as follows:

$$\max_{\boldsymbol{\pi} \in \Pi} J(\boldsymbol{\pi}, \boldsymbol{r}) \tag{1}$$

i.e. find the policy that gives maximum expected discounted reward in the MDP M under initial state distribution ν. Solving the problem 1 is known as *direct policy optimization*. It is a **non-concave** optimization problem but we can find algorithms that converge globally on it [2]. Alternatively because of observation 1.2

(policy-occupancy measure bijection), one can also formulate solving 1 in terms of the occupancy measure as follows:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \tilde{\Omega}(\boldsymbol{\mu}) \\ \text{s.t.} \quad & (E - \gamma P)^T \boldsymbol{\mu} = (1-\gamma)\boldsymbol{\nu}, \end{aligned} \tag{2}$$

where the equality constraints $(E - \gamma P)^T \boldsymbol{\mu} = (1-\gamma)\boldsymbol{\nu}$ (the **Bellman Flow Constraints**) are equivalent to restricting $\boldsymbol{\mu}$ to the occupancy set $\mathcal{M}$ (see definition 1.9). Note that as the problem (2) has a concave cost-function with a set of linear equality constraints, it is a concave program. Observe that in the special case where $\tilde{\Omega}(\boldsymbol{\mu}) = 0$ (2) becomes a Linear Program that can be solved in poly-time. The problem (2), naturally relaxes to a an associated, unconstrained Lagrangian form:

$$\max_{\boldsymbol{\mu}} \min_{\boldsymbol{\lambda} \geq 0} \quad \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \tilde{\Omega}(\boldsymbol{\mu}) + \langle \boldsymbol{\lambda}, \big((E - \gamma P)^T \boldsymbol{\mu} - (1-\gamma)\boldsymbol{\nu} \big) \rangle, \tag{3}$$

in which case the problem becomes a *saddle-point problem*. In the unregularized form ($\tilde{\Omega}(\boldsymbol{\mu}) = 0$) the problem is *bilinear* w.r.t the lagrangian multiplier $\boldsymbol{\lambda}$ and the occupancy measure (primal variable) $\boldsymbol{\mu}$, else it is concave-linear. We sometimes use notation $J(\boldsymbol{\mu}, \boldsymbol{r}) = J(\mathrm{OM}(\boldsymbol{\pi}), \boldsymbol{r})$ and $K(\boldsymbol{\mu},) = K(\mathrm{OM}(\boldsymbol{\pi}))$.

1.2 Constrained Markov Decision Processes

The *Constrained Markov Decision Process* (CMDP) is an extension of the above-defined MDP that incorporates a cost function as well as constraints on that cost function, we define the CMDP as the following 8-tuple:

$$\text{CMDP} = (S, A, P, \boldsymbol{\nu}, \boldsymbol{r}, \Psi, \boldsymbol{b}, \gamma),$$

where the terms $S, A, P, \boldsymbol{\nu}, \boldsymbol{r}, \gamma$ are defined as in an unconstrained MDP (see section 1.1) but we further incorporate a cost function $\Psi : S \times A \to \mathbb{R}^l$ which, in the discrete state-action setting, we can conveniently represent as a matrix $\Psi \in \mathbb{R}^{nm \times l}$ (here l is the constraint-dimensionality). We denote the i-th element of the output of the cost function Ψ: ψ_i. The constraint vector $\boldsymbol{b} \in \mathbb{R}^l$ allows us to define our CMDP constraints.

1.2.1 Discounted Costs, Cost-Value, Cost-Quality, Cost-Objective function

We define equivalent quantities (discounted cost, value, quality and objective) for cost to the ones we defined for the reward function.

Definition 1.13. *(**Discounted Cost**) For a given trajectory τ of the MDP M one can compute the i-th element of the **discounted cost** $U(\tau) \in \mathbb{R}^l$ as follows:*

$$U(\tau, i) = (1-\gamma) \lim_{T \to +\infty} \sum_{T}^{t=0} \gamma^t \psi_i(s_t, a_t),$$

we denote the full vector obtained by stack elements as defined above as $\boldsymbol{U}(\tau)$,

$$\boldsymbol{U}(\tau) = (1-\gamma) \lim_{T \to +\infty} \sum_{T}^{t=0} \gamma^t \Psi(s_t, a_t).$$

Definition 1.14. *(**Cost-Value Function**) For a given policy $\pi : S \to \Delta A$ and under the assumption that the agent starts at some state s_0, we define the **cost-value function** $\boldsymbol{V}_\Psi^\pi : S \to \mathbb{R}^l$ as:*

$$\boldsymbol{V}_\Psi^\pi(s) = \mathbb{E}\Big[\boldsymbol{U}(\tau)\Big| s_0 = s, \pi\Big] = \mathbb{E}\left[(1-\gamma)\sum_{t=0}^{+\infty} \gamma^t \Psi(s_t, a_t)\Big| s_0 = s, \pi\right].$$

Definition 1.15. *(**Cost-Quality Function**) For a given policy $\pi : S \to \Delta A$ and under the assumption that the agent starts at some state s_0 with first action a_0, we define the **cost-quality function** $\boldsymbol{Q}^\pi : S \times A \to \mathbb{R}^l$ as:*

$$\boldsymbol{Q}_\Psi^\pi(s, a) = \mathbb{E}\Big[\boldsymbol{U}(\tau)\Big| s_0 = s, a_0 = a, \pi\Big] = \mathbb{E}\left[(1-\gamma)\sum_{t=0}^{+\infty} \gamma^t \Psi(s_t, a_t)\Big| s_0 = s, a_0 = a, \pi\right].$$

Definition 1.16. *(**Cost-Objective Function**) For a given policy $\pi : S \to \Delta A$ and under an initial state distribution $\boldsymbol{\nu} \in \Delta_S$ we define the **cost-objective function** as:*

$$\boldsymbol{K}(\pi) = \mathbb{E}\Big[\boldsymbol{U}(\tau)\Big| s_0 \sim \nu, \pi\Big] = \mathbb{E}\Big[V_\Psi^\pi(s)\Big| s \sim \nu\Big].$$

Observation 1.3. *As in the unconstrained MDP setting, one can rewrite the cost-objective as a function of the occupancy measure:*

$$\Psi^T \boldsymbol{\mu}^\pi = \boldsymbol{K}(\pi).$$

1.2.2 Feasibility

The main difference between *MDP*s and *CMDP*s is that in a *CMDP*, not all policies are admissible, in order to be feasible, a policy needs to meet the constraint:

$$\boldsymbol{K}(\pi) \leq \boldsymbol{b},$$

or equivalently

$$\Psi^T \boldsymbol{\mu}^\pi \leq \boldsymbol{b}.$$

This restricts the set of admissible policies and occupancy measures and allows us to define the *feasible set* of the CMDP.

Definition 1.17. ***(Feasible Set)*** *Given a CMPD C we define it's* ***feasible set*** $\mathcal{F}$ *as the set of points that satisfy the bellman flow constraints without violating the constraints:*

$$\mathcal{F} := \left\{ \mu \in \mathcal{M} \middle| \Psi^T \mu \leq \boldsymbol{b} \right\}.$$

1.2.3 Goal of the CMDP, formulating optimization problems

The goal of a CMDP can be stated as follows:

$$\begin{aligned} &\max_{\pi \in \Pi} J(\boldsymbol{\pi}, \boldsymbol{r}), \\ &\quad \text{s.t } \boldsymbol{J}_\Psi^\nu(\pi) \leq \boldsymbol{b}, \end{aligned} \tag{4}$$

which in plain english reads as "*find the policy that maximizes the expected discounted reward while keeping the expected discounted cost below the constraints*". As we already did in section 1.1.3, we can reformulate our problems in terms of the occupancy measure:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \boldsymbol{r}^T \boldsymbol{\mu} - \tilde{\Omega}(\mu), \\ \text{s.t.} \quad & (E - \gamma P)^T \boldsymbol{\mu} = (1-\gamma)\boldsymbol{\nu}, \\ & \Psi^T \boldsymbol{\mu} \leq \boldsymbol{b}, \end{aligned} \tag{5}$$

as we showed in section 1.1.3 for the unconstrained setting, this yields a concave program, with linear constraints, which becomes an LP in the special case where $\Omega(\pi) = 0$.

1.2.4 Solution Maps

We formulate the following solution maps are associated with the solutions of the optimization problems associated with the (constrained or unconstrained) MDPs.

Definition 1.18. *(**Policy Solution Map**) Given a CMDP C and it's reward* $\boldsymbol{r}$*, we define it's* ***policy solution map*** *as:*

$$CRL_{\pi}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} J(\boldsymbol{\pi}, \boldsymbol{r})$$
$$s.t.\ OM(\pi) \in \mathcal{F}$$

Definition 1.19. *(**Occupancy Solution Map**) Given a CMDP C and it's reward* $\boldsymbol{r}$*, we define it's* ***occupancy solution map*** *as:*

$$CRL_{\mu}(\boldsymbol{r}) = \arg\max_{\mu \in \mathcal{F}} J(\boldsymbol{\mu}, \boldsymbol{r})$$

Note that we also use the notations $\mathrm{RL}_{\pi/\mu}(\boldsymbol{r})$ for the unconstrained solution maps in occupancy and policy.

Observation 1.4. *The solution maps from def 1.18 and from def 1.19 are equivalent under the bijection from observation 1.2, since by observation 1.1 the optimized values are identical.*

1.3 Inverse Reinforcement Learning

We will now consider the inverse problem to the MDP and CMDP problems that we discussed before, **Constrained Inverse Reinforcement Learning** (IRL). In the IRL setting (*on a known MDP*) we are given an *MDP* or a *CMDP* without its reward:

$$CMDP \setminus \boldsymbol{r} = (S, A, P, \boldsymbol{\nu}, \Psi, \boldsymbol{b}, \gamma),$$

as well as a dataset $\mathcal{D}$ of expert example in the form of trajectories:

$$\mathcal{D} = \{\tau_1, \tau_2, ..., \tau_N\},$$

produced by some expert policy $\boldsymbol{\pi}^E$. Assuming the dataset is large enough (which we will do at first), this is equivalent to getting an approximation of the expert's occupancy measure $\boldsymbol{\mu}^E$. In the CIRL setting we also generally restrict the reward class $\mathcal{R}$ to some arbitrary convex set in $\mathbb{R}^{nm}$.

Definition 1.20. *(**CIRL Solution Map**) The **goal of CIRL** is to find some mapping $CIRL_\pi : \Pi \to \mathcal{R}$ (or alternatively $CIRL_\mu : \mathcal{M} \to \mathcal{R}$). That satisfies*

$$(CRL_\pi \circ CIRL_\pi)(\boldsymbol{\pi}^E) = \boldsymbol{\pi}^E.$$

In plain english **we want to find a method for recovering a reward for which the original expert is optimal.**

Assumption 1.1. *(**Realizability**) We assume that the expert policy is optimal w.r.t. some reward $\boldsymbol{r}^E \in \mathcal{R}$, i.e.*

$$\boldsymbol{\mu}^E = CRL_\mu(\boldsymbol{r}^E).$$

The goal of the IRL problem is to recover a reward $\hat{\boldsymbol{r}}$ s.t. the expert $\boldsymbol{\mu}^E = \mathrm{CRL}_\mu(\hat{\boldsymbol{r}})$ (the expert is optimal for that reward).

Proposition 1.1. *(**Min-Max Program to solve IRL**) If assumption 1.1 is true, then the rewards optimizing the program:*

$$\min_{\boldsymbol{r} \in \mathcal{R}} \max_{\boldsymbol{\mu} \in \mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\mu} - \boldsymbol{\mu}^E \rangle - \tilde{\Omega}(\boldsymbol{\mu}), \tag{6}$$

are exactly the rewards in $\mathcal{R}$ for which $\boldsymbol{\mu}^E$ is optimal.

Proof. (Of proposition 1.1) We can rewrite an equivalent problem to (6) as follows:

$$\min_{\boldsymbol{r} \in \mathcal{R}} \max_{\boldsymbol{\mu} \in \mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \tilde{\Omega}(\boldsymbol{\mu}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \tilde{\Omega}(\boldsymbol{\mu}^E),$$

where the addition of the $\tilde{\Omega}(\boldsymbol{\mu}^E)$ term has no influence on the optimizers, as it is a function of neither of the decision variables. For convenience we write:

$$\boldsymbol{r}^T \boldsymbol{\mu} - \tilde{\Omega}(\boldsymbol{\mu}) - \boldsymbol{r}^T \boldsymbol{\mu}^E + \tilde{\Omega}(\boldsymbol{\mu}^E) = L(\boldsymbol{r}, \boldsymbol{\mu}).$$

For any fixed $\boldsymbol{r} \in \mathcal{R}$ it holds that

$$\arg\max_{\boldsymbol{\mu} \in \mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \tilde{\Omega}(\boldsymbol{\mu}) - \overbrace{\langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \tilde{\Omega}(\boldsymbol{\mu}^E)}^{\text{constant w.r.t } \boldsymbol{\mu}} = \mathrm{CRL}_\mu(\boldsymbol{r}),\ \boldsymbol{r} \in \mathcal{R},$$

i.e. the optimal $\boldsymbol{\mu}$ for any fixed $\boldsymbol{r} \in \mathcal{R}$ gives the optimal policy. Moreover, for any fixed $\boldsymbol{r} \in \mathcal{R}$, we get that:

$$\max_{\boldsymbol{\mu} \in \mathcal{F}} L(\boldsymbol{r}, \boldsymbol{\mu}). \geq 0,\ \boldsymbol{r} \in \mathcal{R},$$

since:

1. either μ^E maximizes the value $\langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle - \tilde{\Omega}(\boldsymbol{\mu}^E)$, in which case, we can set the optimizer $\boldsymbol{\mu} = \boldsymbol{\mu}^E$ and get $L(\boldsymbol{r}, \boldsymbol{\mu}) = 0$,
2. or μ^E does not maximize the value $\langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle - \tilde{\Omega}(\boldsymbol{\mu}^E)$, in which case we get $L(\boldsymbol{r}, \boldsymbol{\mu}) > 0$.

Observe that the lower bound ($L(\boldsymbol{r}, \boldsymbol{\mu}) = 0$) is only achieved if $\langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle - \tilde{\Omega}(\boldsymbol{\mu}^E) = \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \tilde{\Omega}(\boldsymbol{\mu})$, i.e. if $\boldsymbol{\mu}^E \in \arg\max_{\boldsymbol{\mu} \in \mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \tilde{\Omega}(\boldsymbol{\mu}^E)$ which is equivalent to $\boldsymbol{\mu}^E \in \mathrm{CRL}_\mu(\boldsymbol{r})$. I.e. the optimal reward $\boldsymbol{r}^*$ of the min-max problem is such that the expert $\boldsymbol{\mu}^E$ is optimal w.r.t $\boldsymbol{r}^*$.

□

2 Min-Max Optimization and Saddle Point Problems

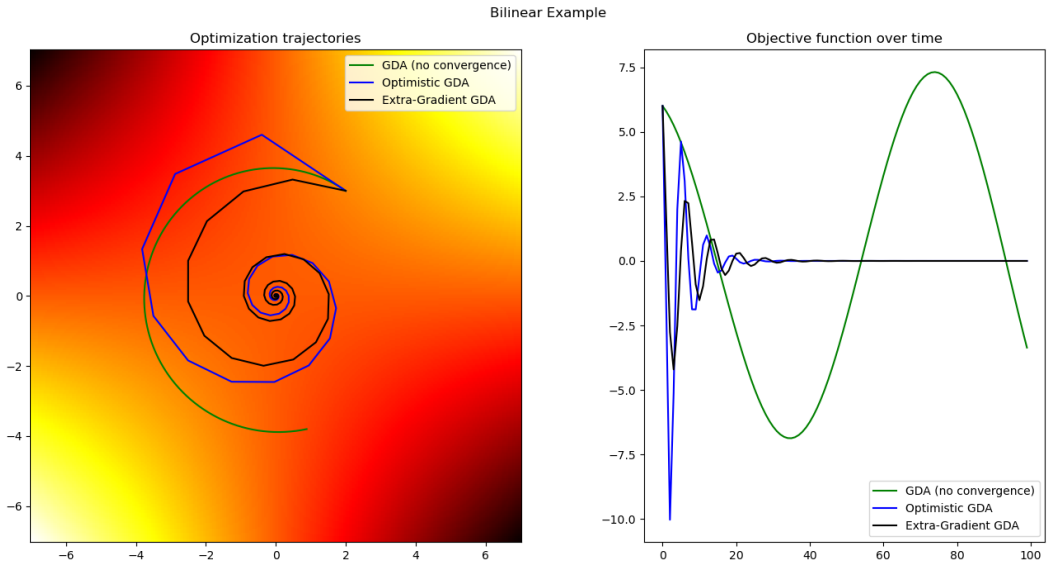

Figure 1: Iterations of three optimization methods (Gradient Descent-Ascent, Optimistic Gradient and Extra Gradient methods) on an unconstrained bilinear saddle-point optimization problem.

In the following section, we introduce, discuss and analyze algorithms for solving *saddle-point* problems of the form;

$$\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x},\boldsymbol{y}).$$

Where $f : X \times Y \to \mathbb{R}$ is some scalar function for which we want to find a *saddle point*, i.e. a point $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in X \times Y$, s.t.

$$f(\boldsymbol{x}^*, \boldsymbol{y}) \leq f(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq f(\boldsymbol{x}, \boldsymbol{y}^*),$$

i.e. in plain english "$(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is a minimizer in* $\boldsymbol{x}$ *and a maximizer in* $\boldsymbol{y}$". Such problems which will prove especially relevant in our study of inverse reinforcement learning.

2.1 Preliminaries

Definition 2.1. ***(Lipschitz-continuous function)*** *A function* $f : \boldsymbol{dom}(f) \to \mathbb{R}^m$ *is* B*-Lipschitz if for any* $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f)$*, there exists* $B \in \mathbb{R}_+$

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leq B\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Definition 2.2. ***(Smooth function)*** *A function* $f : \boldsymbol{dom}(f) \to \mathbb{R}$ *is called* L*-smooth if it has* L*-Lipschitz continuous gradients, i.e. if for any* $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f)$*, there exists* $L \in \mathbb{R}_+$*:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Definition 2.3. *(**Convex set**) A set $C \subseteq \mathbb{R}^d$ is convex if for any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the connecting line segment is contained in C, i.e. $\forall \lambda, \lambda \in [0,1]$:*

$$\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in C.$$

Definition 2.4. *(**Convex function**) A function $f : \boldsymbol{dom}(f) \to \mathbb{R}$ is called convex on $\boldsymbol{dom}(f)$ if*

1. *$\boldsymbol{dom}(f)$ is convex,*
2. *for all $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f)$ and $\lambda \in [0,1]$ we have:*

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$

Definition 2.5. *(**Concave function**) A function $f : \boldsymbol{dom}(f) \to \mathbb{R}$ is called concave on $\boldsymbol{dom}(f)$ if the function $g : \boldsymbol{dom}(f) \to \mathbb{R}$ defined as $g(x) = -f(x)$ is convex on $\boldsymbol{dom}(f)$.*

Definition 2.6. *(**Saddle point**) The point $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in X \times Y$ is a saddle point of the function $f : X, Y \to \mathbb{R}$ if for any $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in Y$ we have:*

$$f(\boldsymbol{x}^*, \boldsymbol{y}) \leq f(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq f(\boldsymbol{x}, \boldsymbol{y}^*).$$

Definition 2.7. *(**Saddle-Point Problem**) Consider $f : X, Y \to \mathbb{R}$ a continuously differentiable scalar function on convex domains $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$. For convenience use notation $Z = X \times Y$ and $z = (x, y)$. We call the optimization problem below:*

$$\min_{\boldsymbol{x} \in X} \max_{\boldsymbol{y} \in Y} f(\boldsymbol{x}, \boldsymbol{y}).$$

a saddle point problem. The solution set $\mathcal{Z}$ defined as:

$$\mathcal{Z} := \left\{ (x^*, y^*) \in X \times Y \,\middle|\, f(\boldsymbol{x}^*, \boldsymbol{y}) \leq f(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq f(\boldsymbol{x}, \boldsymbol{y}^*),\ \forall x \in X,\ y \in Y \right\}$$

is the set of all saddle points (see def 2.6) of the function f. Such a problem can be constrained ($X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^m$) or unconstrained ($X = \mathbb{R}^n$, $Y = \mathbb{R}^m$).

Definition 2.8. *(**Convex-Concave Problem**) Consider a saddle point problem (see def 2.7), defined by $f : X, Y \to \mathbb{R}$ a continuously differentiable scalar function on convex domains $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$. We call the problem a convex-concave problem if f is convex in x and concave in y.*

Proposition 2.1. *In general:*

$$\max_{\boldsymbol{x} \in X} \min_{\boldsymbol{y} \in Y} f(\boldsymbol{x}, \boldsymbol{y}) \leq \min_{\boldsymbol{y} \in Y} \max_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \boldsymbol{y}).$$

Proof. Let $\tilde{\boldsymbol{x}} = \arg\max_{\boldsymbol{y} \in Y} \min_{\boldsymbol{x} \in X} f(x, y)$, $\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y} \in Y} \min_{\boldsymbol{x} \in X} f(x, y)$, $\square$

Theorem 2.1. *(**Sion's Minimax Theorem** [1]) If X and Y are convex compact sets, and if $f : X \times Y \to \mathbb{R}$ is convex concave, then:*

$$\max_{\boldsymbol{x} \in X} \min_{\boldsymbol{y} \in Y} f(\boldsymbol{x}, \boldsymbol{y}) = \min_{\boldsymbol{y} \in Y} \max_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \boldsymbol{y}).$$

Definition 2.9. *(**Projection Operator**) given some convex set $S \subseteq \mathbb{R}^n$ as well a point $\boldsymbol{x} \in \mathbb{R}^n$, we define the projection $\boldsymbol{p}$ of $\boldsymbol{x}$ onto S as:*

$$\boldsymbol{p} = \arg\min_{\boldsymbol{y} \in S} \|\boldsymbol{x} - \boldsymbol{y}\|,$$

which we often simply denote as the output of the projection operator associated with the set S:

$$\Pi_S(\boldsymbol{x}) := \arg\min_{\boldsymbol{y} \in S} \|\boldsymbol{x} - \boldsymbol{y}\|.$$

When the norm used is the Euclidian norm (or L_2 norm), we call this operation the Euclidian Projection.

Definition 2.10. *(**Duality Gap**) We define the duality gap as a way of characterizing the sub-optimality of a point. Given some point $\tilde{\boldsymbol{z}} = (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in Z$, its duality gap is given by:*

$$\textit{Duality Gap}(\tilde{\boldsymbol{z}}) := \max_{\boldsymbol{y} \in Y} f(\tilde{\boldsymbol{x}}, \boldsymbol{y}) - \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \tilde{\boldsymbol{y}})$$

2.2 Gradient Descent Ascent

The most simple algorithm is the natural extension of gradient descent/ascent to the saddle point problem. In the following section, we define the gradient descent-ascent (GDA) algorithm, analyze it and discuss it's performance and limitations. Note that we specifically analyze *projected* gradient descent-ascent, as results for projected GDA trivially generalize to unconstrained GDA and the projected setting will be useful for our applications in inverse reinforcement learning.

Algorithm 1: (Projected) Gradient Descent Ascent

Set the learning rate $\eta > 0$
Initialize the algorithm at some point $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
foreach *iteration* $k = 0, 2, ..., K-1$ **do**
  $\boldsymbol{x}_{k+1} \leftarrow \Pi_X\big(\boldsymbol{x}_k - \eta \nabla_x f(x_k, y_k)\big)$
  $\boldsymbol{y}_{k+1} \leftarrow \Pi_Y\big(\boldsymbol{y}_k + \eta \nabla_y f(x_k, y_k)\big)$
end
return $(\boldsymbol{x}_N, \boldsymbol{y}_N)$

Proposition 2.2. *($O(\frac{1}{\sqrt{K}})$ **convergence rate**) For B-Lipschitz convex concave functions $f : X, Y \to \mathbb{R}$, defined on two convex domains X and Y with bounded diameter $D = \max\{diam(X), diam(Y)\}$, after K steps, projected gradient descent-ascent (alg 1) with learning rate $\eta_k = \frac{D}{\sqrt{2kB}}$ has gives an average point for which the duality gap is bounded by:*

$$\textit{Duality Gap}(\bar{\boldsymbol{z}}_K) \leq \frac{4DB}{\sqrt{K}},$$

where $\bar{\boldsymbol{z}}_K = \frac{1}{K} \sum_{k=1...K} (\boldsymbol{x}_k, \boldsymbol{y}_k)$.

Proof. (of proposition 2.2) Given a fixed-point $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$, using the projected gradient update on $\boldsymbol{x}$ we get:

$$\begin{aligned}
\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 &= \|\Pi_X(\boldsymbol{x}_k - \eta_k \nabla_x f(\boldsymbol{x}_k, \boldsymbol{y}_k)) - \Pi(\boldsymbol{x}^*)\|^2 && \text{since } \boldsymbol{x}^* \in X \\
&\leq \|\boldsymbol{x}_k - \eta_k \nabla_x f(\boldsymbol{x}_k, \boldsymbol{y}_k) - \boldsymbol{x}^*\|^2 && \text{projections are non-expansive} \\
&= \|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \eta_k^2 \|\nabla_x f(\boldsymbol{x}_k, \boldsymbol{y}_k)\|^2 \\
&\quad - 2\eta_K \langle \boldsymbol{x}_k - \boldsymbol{x}^*, \nabla_x f(\boldsymbol{x}_k, \boldsymbol{y}_k) \rangle \\
&\leq \|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \eta_k^2 B^2 && \text{since } f \text{ is } B\text{-Lipschitz} \\
&\quad - 2\eta_K [f(\boldsymbol{x}_k, \boldsymbol{y}_k) - f(\boldsymbol{x}^*, \boldsymbol{y}_k)] && \text{by convexity of } f
\end{aligned}$$

And thus:

$$\Rightarrow \quad f(\boldsymbol{x}_k, \boldsymbol{y}_k) - f(\boldsymbol{x}^*, \boldsymbol{y}_k) \leq \frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2}{2\eta_K} + \frac{\eta_K}{2} B^2. \qquad (a)$$

Similarly using the projected gradient update on $\boldsymbol{y}$ we have:

$$\begin{aligned}
\|\boldsymbol{y}_{k+1} - \boldsymbol{y}^*\|^2 &\leq \|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 + \eta_k^2 \|\nabla_y f(\boldsymbol{x}_k, \boldsymbol{y}_k)\|^2 \\
&\quad + 2\eta_K \langle \boldsymbol{y}_k - \boldsymbol{y}^*, \nabla_y f(\boldsymbol{x}_k, \boldsymbol{y}_k) \rangle \\
&\leq \|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 + \eta_k^2 B^2 && \text{since } f \text{ is } B\text{-Lipschitz} \\
&\quad - 2\eta_K [f(\boldsymbol{x}_k, \boldsymbol{y}_k) - f(\boldsymbol{x}_k, \boldsymbol{y}^*)] && \text{by concavity of } f
\end{aligned}$$

And thus:

$$\Rightarrow \quad f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}_k, \boldsymbol{y}_k) \leq \frac{\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 - \|\boldsymbol{y}_{k+1} - \boldsymbol{y}^*\|^2}{2\eta_K} + \frac{\eta_K}{2} B^2. \qquad (b)$$

Adding (a) and (b) together we get:

$$\begin{aligned}
&f(\boldsymbol{x}_k, \boldsymbol{y}_k) - f(\boldsymbol{x}^*, \boldsymbol{y}_k) + f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}_k, \boldsymbol{y}_k) = f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}^*, \boldsymbol{y}_k) \\
&\leq \frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2}{2\eta_K} + \frac{\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 - \|\boldsymbol{y}_{k+1} - \boldsymbol{y}^*\|^2}{2\eta_K} + \eta_K B^2. \qquad (c)
\end{aligned}$$

Summing up over all optimization steps we have:

$$\begin{aligned}
&\sum_{k=1\ldots K} f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}^*, \boldsymbol{y}_k) \\
&\leq \sum_{k=1\ldots K} \left[\frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2}{2\eta_K} \right] \\
&+ \sum_{k=1\ldots K} \left[\frac{\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 - \|\boldsymbol{y}_{k+1} - \boldsymbol{y}^*\|^2}{2\eta_K} \right] + B^2 \sum_{k=1\ldots K} \eta_K. \qquad (d)
\end{aligned}$$

Observing that the terms in the sum telescope we have:

$$\begin{aligned}
\sum_{k=1\ldots K} \left[\frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2}{2\eta_K} \right] &\leq \frac{D^2}{2\eta_K} \\
\sum_{k=1\ldots K} \left[\frac{\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2 - \|\boldsymbol{y}_{k+1} - \boldsymbol{y}^*\|^2}{2\eta_K} \right] &\leq \frac{D^2}{2\eta_K}.
\end{aligned}$$

Which once plugged into (d) give:

$$\sum_{k=1\ldots K} f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}^*, \boldsymbol{y}_k) \leq \frac{D^2}{\eta_K} + B^2 \sum_{k=1\ldots K} \eta_k = \frac{D^2\sqrt{K}}{\eta} + B^2\eta \sum_{k=1\ldots K} \frac{1}{\sqrt{k}}$$
$$\leq \frac{D^2\sqrt{K}}{\eta} + 2B^2\eta\sqrt{K}.$$

Thus we have:

$$\frac{1}{K}\Big[\sum_{k=1\ldots K} f(\boldsymbol{x}_k, \boldsymbol{y}^*) - f(\boldsymbol{x}^*, \boldsymbol{y}_k)\Big] \leq \frac{D^2}{\eta\sqrt{K}} + 2B^2\eta\frac{1}{\sqrt{K}}$$
$$\text{(Optimizing } \eta) \quad = \frac{4DB}{\sqrt{K}}$$

□

2.3 Extra-Gradient Descent Ascent

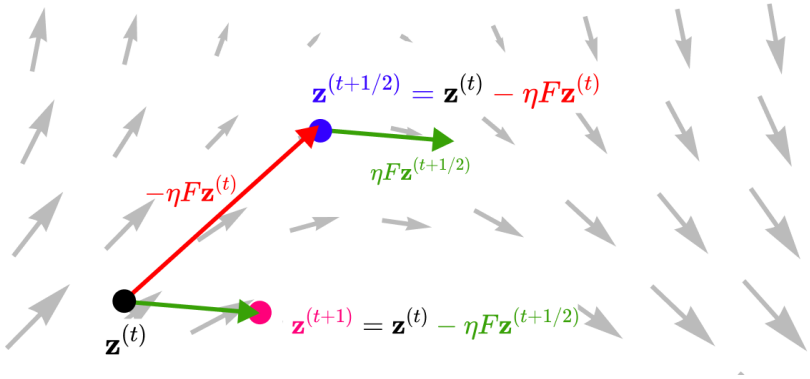

Figure 2: An illustration of an extra-gradient step.

Next, we consider the Extra-Gradient Descent Ascent method (EG), which provides an approximation of the *implicit* proximal point method and has better performance compared to the naïve GDA approach, specifically:

1. a better convergence rate on general convex-concave programs ($O(\frac{1}{K})$ instead of $O(\frac{1}{\sqrt{K}})$)
2. last iterate convergence for all convex-concave functions, including for bilinear functions (which isn't true for naïve GDA).

As in the naïve GDA case, we choose to analyze *projected* EG, because this will

prove useful to solve the IRL problems we are concerned about.

Algorithm 2: (Projected) Extra-Gradient Descent Ascent

Set the learning rate $\eta > 0$
Initialize the algorithm at some point $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
foreach *iteration* $k = 0, 2, ..., K-1$ **do**
  $\boldsymbol{x}_{k+1/2} \leftarrow \Pi_X\big(\boldsymbol{x}_k - \eta \nabla_x f(\boldsymbol{x}_k, \boldsymbol{y}_k)\big)$
  $\boldsymbol{y}_{k+1/2} \leftarrow \Pi_Y\big(\boldsymbol{y}_k + \eta \nabla_y f(\boldsymbol{x}_k, \boldsymbol{y}_k)\big)$
  $\boldsymbol{x}_{k+1} \leftarrow \Pi_X\big(\boldsymbol{x}_k - \eta \nabla_x f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2})\big)$
  $\boldsymbol{y}_{k+1} \leftarrow \Pi_Y\big(\boldsymbol{y}_k + \eta \nabla_y f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2})\big)$
end
return $(\boldsymbol{x}_K, \boldsymbol{y}_K)$

In order to make the computations a bit lighter, we will use the following notation, let F denote the flow (the direction of gradients ascending and descending) of GDA/EG:

$$F(\boldsymbol{z}) = \begin{bmatrix} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) \\ -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}). \end{bmatrix}$$

We also define a single the feasible set $S := \{\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}); \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$, and it's associated projection operator $\Pi_S(\boldsymbol{z}) = [\Pi_X(\boldsymbol{x}), \Pi_Y(\boldsymbol{y})]^T$. In this setting we can express the EG updates, in a much more concise form as:

$$\begin{aligned} \boldsymbol{z}_{k+1/2} &= \Pi_S(\boldsymbol{z}_k - \eta F(\boldsymbol{z}_k)), \\ \boldsymbol{z}_{k+1} &= \Pi_S(\boldsymbol{z}_k - \eta F(\boldsymbol{z}_{k+1/2})). \end{aligned}$$

Given a fixed point $\boldsymbol{z}^*$, and using the first order characterization of optimality we have that $\forall (\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$:

$$\begin{aligned} \langle \nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}), \boldsymbol{x} - \boldsymbol{x}^* \rangle \geq 0, \\ \langle -\nabla_y f(\boldsymbol{x}, \boldsymbol{y}^*), \boldsymbol{y} - \boldsymbol{y}^* \rangle \geq 0. \end{aligned}$$

which can be expressed more concisely as:

$$\left\langle \begin{bmatrix} \nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}) \\ -\nabla_y f(\boldsymbol{x}, \boldsymbol{y}^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}^* \end{bmatrix} - \begin{bmatrix} \boldsymbol{y}^* \\ \boldsymbol{y}^* \end{bmatrix} \right\rangle \geq 0 \Rightarrow \langle F(\boldsymbol{z}^*), \boldsymbol{z} - \boldsymbol{z}^* \rangle \geq 0.$$

Observe that for ∇f μ-Lipschitz, F is a 2μ-Lipschitz operator.

Proposition 2.3. *For μ-smooth convex-concave functions, where X and Y have max diameter D, expected gradient with $\eta_k = \frac{1}{2\mu}$ gives the following duality gap for the expected steps: the duality gap is bounded by:*

$$\textit{Duality Gap}(\bar{\boldsymbol{z}}_K) \leq \frac{2D^2\mu}{K},$$

where $\bar{\boldsymbol{z}}_K = \frac{1}{K} \sum_{k=1...K} (\boldsymbol{x}_k, \boldsymbol{y}_k)$.

Proof. For some $\boldsymbol{x} \in X$, $\boldsymbol{y} \in Y$, we have:

$$
\begin{aligned}
& f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}) - f(\boldsymbol{x}, \boldsymbol{y}_{k+1/2}) \\
&= f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}) \overbrace{-f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2}) + f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2})}^{=0} - f(\boldsymbol{x}, \boldsymbol{y}_{k+1/2}) \\
& \qquad \text{(by convexity/concavity)} \\
&\leq \langle \nabla_{\boldsymbol{y}} f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2}), \boldsymbol{y} - \boldsymbol{y}_{k+1/2} \rangle + \langle \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2}), \boldsymbol{x}_{k+1/2} - \boldsymbol{x} \rangle \\
&= \left\langle \begin{bmatrix} \nabla_x f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2}) \\ -\nabla_y f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}_{k+1/2}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{x}_{k+1/2} - \boldsymbol{x} \\ \boldsymbol{y}_{k+1/2} - \boldsymbol{y} \end{bmatrix} \right\rangle \\
&\Rightarrow f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}) - f(\boldsymbol{x}, \boldsymbol{y}_{k+1/2}) \leq \langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle.
\end{aligned}
$$

This provides us with a way to bound $\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle$ and thus the duality gap, by finding some upper-bound for the right-hand side of the equation, $\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle$, which is what we will do next.

$$
\begin{aligned}
\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle &= \left\langle \frac{\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \right\rangle \\
& \qquad \text{(From the upddates, the tilde means before projection.)} \\
&= \left\langle \frac{\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle + \left\langle \frac{\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \right\rangle \\
& \qquad \text{Adding } 0 = \boldsymbol{z}_{k+1} - \mathbf{z}_{k+1} \\
&\leq \left\langle \frac{\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle + \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1}}{\eta}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \right\rangle \\
& \text{(Using that } \langle \tilde{\boldsymbol{z}}_{k+1} - \boldsymbol{z}_{k+1}, \boldsymbol{z} - \boldsymbol{z}_{k+1} \rangle \Rightarrow \langle -\tilde{\boldsymbol{z}}_{k+1}, \boldsymbol{z} - \boldsymbol{z}_{k+1} \rangle \leq -\boldsymbol{z}_{k+1}, \boldsymbol{z} - \boldsymbol{z}_{k+1} \rangle \\
& \qquad \text{from the properties of the projection operator.)} \\
&= \left\langle \frac{\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1/2}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle + \left\langle \frac{\tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle \\
& \qquad + \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1}}{\eta}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \right\rangle \\
& \qquad \text{Adding } 0 = \tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1/2} \\
&= \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle + \left\langle \frac{\tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle \\
& \qquad + \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1}}{\eta}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \right\rangle \\
& \text{(Using that } \langle \tilde{\boldsymbol{z}}_{k+1/2} - \boldsymbol{z}_{k+1/2}, \boldsymbol{z} - \boldsymbol{z}_{k+1/2} \rangle \Rightarrow \langle -\tilde{\boldsymbol{z}}_{k+1/2}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle \leq -\boldsymbol{z}_{k+1/2}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \rangle \\
& \qquad \text{from the properties of the projection operator.)}
\end{aligned}
$$

We thus have an expression made up of three terms:

$$\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z}\rangle \leq \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle +$$
$$\left\langle \frac{\tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}}{\eta}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1} \right\rangle + \left\langle \frac{\boldsymbol{z}_k - \boldsymbol{z}_{k+1}}{\eta}, \boldsymbol{z}_{k+1} - \boldsymbol{z} \right\rangle$$
$$\Rightarrow \eta\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z}\rangle \leq \overbrace{\langle \boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle}^{(a)} +$$
$$\overbrace{\langle \tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle}^{(b)} + \overbrace{\langle \boldsymbol{z}_k - \boldsymbol{z}_{k+1}, \boldsymbol{z}_{k+1} - \boldsymbol{z}\rangle}^{(c)}.$$

We will bound these terms individually, starting with (b):

$$\begin{aligned}
&\langle \tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle \\
&= \langle \tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_k, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle + \langle \boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k+1}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle \qquad (\text{add } 0 = \boldsymbol{z}_k - \boldsymbol{z}_k) \\
&= \eta\langle F(\boldsymbol{z}_{k+1/2}) - F(\boldsymbol{z}_k), \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle \qquad (\text{from the updates}) \\
&\leq \eta\|F(\boldsymbol{z}_{k+1/2}) - F(\boldsymbol{z}_k)\| \cdot \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\| \qquad (\text{Cauchy Schwartz}) \\
&\leq 2\eta\mu\|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_k\| \cdot \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\| \qquad (F \text{ is } 2\mu\text{-Lipschitz}) \\
&\leq \frac{1}{2}(4\eta^2\mu^2\|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_k\|^2 \cdot \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\|^2) \qquad (\text{Young's inequality})
\end{aligned}$$

Next, for (a) and (c) we will use that $\langle a, b\rangle = \frac{\|a+b\|^2 - \|a\|^2 - \|b\|^2}{2}$. For (a) we get:

$$\begin{aligned}
&\langle \boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle \\
&= \frac{1}{2}\Big[\|\boldsymbol{z}_k - \boldsymbol{z}_{k+1}\|^2 - \|\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}\|^2 - \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\|^2\Big],
\end{aligned}$$

and for (c) we have:

$$\begin{aligned}
&\langle \boldsymbol{z}_k - \boldsymbol{z}_{k+1}, \boldsymbol{z}_{k+1} - \boldsymbol{z}\rangle \\
&= \frac{1}{2}\Big[\|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_k - \boldsymbol{z}_{k+1}\|^2 - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|^2\Big].
\end{aligned}$$

Putting the three simplified terms together we have:

$$\begin{aligned}
&2\langle \tilde{\boldsymbol{z}}_{k+1/2} - \tilde{\boldsymbol{z}}_{k+1}, \boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\rangle \\
&\leq \Big[\|\boldsymbol{z}_k - \boldsymbol{z}_{k+1}\|^2 - \|\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}\|^2 - \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\|\Big] \\
&\quad + (4\eta^2\mu^2\|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_k\|^2 \cdot \|\boldsymbol{z}_{k+1/2} - \boldsymbol{z}_{k+1}\|^2) \\
&\quad + \Big[\|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_k - \boldsymbol{z}_{k+1}\| - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|\Big] \\
&= \|\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}\|^2(4\mu^2\eta^2 - 1) + \|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|^2.
\end{aligned}$$

Which implies:

$$\begin{aligned}
&\eta\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z}\rangle \\
&\leq \frac{1}{2}\Big[\|\boldsymbol{z}_k - \boldsymbol{z}_{k+1/2}\|^2(4\mu^2\eta^2 - 1) + \|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|^2.\Big]
\end{aligned}$$

Setting, $\eta = \frac{1}{2\mu}$ (optimizing η), we have:

$$\langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle \leq \mu \Big[\|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|^2 \Big].$$

Summing over all ks we thus get:

$$\begin{aligned} \sum_{k=1}^{K} \langle F(\boldsymbol{z}_{k+1/2}), \boldsymbol{z}_{k+1/2} - \boldsymbol{z} \rangle &\leq \sum_{k=1}^{K} \mu \Big[\|\boldsymbol{z}_k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}_{k+1} - \boldsymbol{z}\|^2 \Big] \\ &\leq \mu \|\boldsymbol{z}_1 - \boldsymbol{z}\|^2 \leq 2D^2 \mu. \end{aligned}$$

Now going back to the "top" of the proof, we get our final bound:

$$\frac{1}{K} \sum_{k=1}^{K} f(\boldsymbol{x}_{k+1/2}, \boldsymbol{y}) - f(\boldsymbol{x}, \boldsymbol{y}_{k+1/2}) \leq \frac{\mu \|\boldsymbol{z}_1 - \boldsymbol{z}\|^2 \leq 2D^2 \mu}{K}.$$

Which, since the functions are convex and we can use Jensen's inequality leads to :

$$f(\bar{\boldsymbol{x}}_K, \boldsymbol{y}) - f(\boldsymbol{x}, \bar{\boldsymbol{y}}_K) \leq \frac{\mu \|\boldsymbol{z}_1 - \boldsymbol{z}\|^2 \leq 2D^2 \mu}{K}.$$

Which bounds the duality gap and completes the proof. □

2.4 Optimistic Gradient Descent Ascent

Here we consider the Optimistic Gradient Descent-Ascent method (OG) (sometimes also refered to as Past-Extra Gradient), which provides an alternate approximation of the (implicit) proximal point method. It displays better performance compared to the naïve GDA approach (from section 2.2).

Algorithm 3: (Projected) Optimistic Gradient Descent Ascent

Set the learning rate $\eta > 0$
Initialize the algorithm at some point $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
Initialize the algorithm at some point $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
foreach *iteration* $k = 0, 2, ..., K-1$ **do**
  $\boldsymbol{x}_{k+1} \leftarrow \Pi_X \big(\boldsymbol{x}_k - 2\eta \nabla_x f(x_k, y_k) + \eta \nabla_x f(x_{k-1}, y_{k-1}) \big)$
  $\boldsymbol{y}_{k+1} \leftarrow \Pi_Y \big(\boldsymbol{y}_k + 2\eta \nabla_y f(x_k, y_k) - \eta \nabla_y f(x_{k-1}, y_{k-1}) \big)$
end
return $(\boldsymbol{x}_N, \boldsymbol{y}_N)$

2.5 Summary of Known Results

Below we provide a summary of known results for convergence of gradient-based methods on saddle point problems. Table 1 gives a summary of convergence rates for Lipschitz, smooth convex-concave functions under convex constraints (which roughly corresponds to the occupancy measure formulation of the IRL problem described in section 1.3).

Method	Average-iterate convergence rate	Last-iterate convergence rate
GDA (alg. 1)	$O(1/\sqrt{K})$ (proof of prop.2.2)	No conv. guarantees ($\exists$ limit cycles).
EG (alg. 2)	$O(1/K)$ (proof of prop.2.3)	$O(1/\sqrt{K})$ (Cai et al. 2022 [4])
OG (alg. 3)	$O(1/K)$ (Mokhtari et al. 2020 [3])	$O(1/\sqrt{K})$ (Gorbunov et al. 2022) [5]

Table 1: Convergence results for minimax optimization methods with convex constraints in the general (smooth) convex-concave setting.

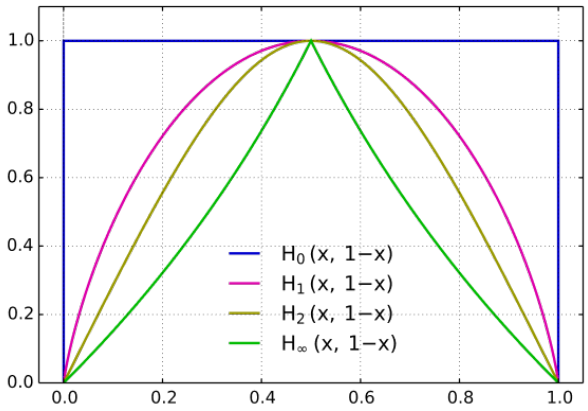

Figure 3: A comparison of Rényi entropies of different orders, plotted as they evolve on a 2-dimensional probability vector. Here we use the result that $\lim \alpha \to 1 H_\alpha = H$ and let the Shannon entropy be denoted by the "first-order Rényi entropy", which is here denoted as H_1.

3 Properties of regularizers

3.1 Shannon Entropy

First, we consider the most common measure of information (or "surpise"), the Shannon Entropy.

Definition 3.1. *(**Shannon Entropy**) For a discrete random variable X, the Shannon Entropy is defined as:*

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i).$$

Observation 3.1. *(**Properties of the Shannon Entropy**) The Shannon entropy has the nice property of being a (strictly) concave function, but regardless, it poses several problems for first-order optimization methods. One big problem is that H is not defined when a probability in the distribution goes to 0 (as $\log 0$ does not exist), as $\lim_{x\to 0} x \log x = 0$ we can for convenience re-define H as :*

$$H(X) := \begin{cases} -\sum_{i=1}^{n} p_i \log(p_i) & \text{if } p_1 > 0 \forall i \in [n], \\ 0 & \text{otherwise.} \end{cases}$$

This approach "plugs" the hole in the Shannon entropy but doesn't fix all of the problems associated with. Specifically, the gradient of H, which is expressed as:

$$[\nabla H(X)]_i = -\log p_i - 1,$$

is neither Lipschitz nor bounded, this is easily verifiable by observing that since $\lim_{x\to 0a} \log(x) = -\infty$*, when we pick a probability vector* $\boldsymbol{p}$*, s.t. for which some element* $p_i \to 0$ *we have that:* $\|\nabla H(\boldsymbol{p})\| \to +\infty$*. This makes* H *violate an essential requirement for convergence of first-order optimization methods. (The gradient does have the big advantage of being separable coordinate, by coordinate, which will prove useful later.)*

3.1.1 Shannon Shenanigans: Lipschitzness of the Shannon Entropy over $\Delta_n^{(\rho)}$

The following section is sort of a detour around the Shannon Entropy properties. The fact that H is non Lipschitz when some probability in the distribution goes to 0 motivates the study of the properties of H on a restricted domain, for which we guarantee that no probability p_i ever goes to close to 0, for this reason we define the ρ-non-vanishing simplex of dimension n: $\Delta_n^{(\rho)}$.

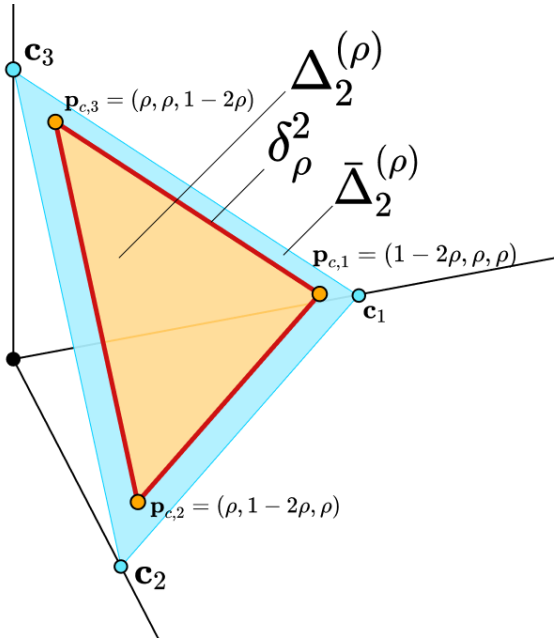

Figure 4: An illustration of the 2 dimensional ρ non-vanishing simplex $\Delta_n^{(\rho)}$ (the orange inner-surface, on the plot). We have that $\Delta_n^{(\rho)} \subset \Delta^2$ (the non-vanishing simplex is a subset of the simplex) and we denote on the plot the area of the simplex Δ^2 that is excluded by $\Delta_n^{(\rho)}$ (i.e. $\bar{\Delta}_\rho^2 = \Delta^2 \setminus \Delta_n^{(\rho)}$) in blue. The plot also labels the corners $\boldsymbol{c}_i$ of the simplex, those $\boldsymbol{p}_{c,i}$ of the non-vanishing simplex, as well as the border $\delta_2^{(\rho)}$ of $\Delta_n^{(\rho)}$ (the red line surrounding the orange area).

Definition 3.2. ***(ρ-non-vanishing Simplex)*** *We define "ρ-non-vanishing simplex"* $\Delta_n^{(\rho)}$*, for some* $0 < \rho < 1/(n+1)$ *as follows :*

$$\Delta_n^{(\rho)} := \Big\{ p \in \Delta \subset \mathbb{R}^n;\ [p]_i > \rho,\ \forall i \in [n+1] \Big\},$$

or equivalently as a convex combination over "corners" $\boldsymbol{p}_{c,i}$ of the ρ non-vanishing simplex:

$$\Delta_n^{(\rho)} = \Big\{ \sum_{i\in[n+1]} \eta_i \boldsymbol{p}_{c,i}, \quad \sum_{i\in[n+1]} \eta_i = 1\},$$

(which will be useful when considering its border). (For a better intuition of what these spaces and objects are, refer to fig. 4) Note that the corner $\boldsymbol{p}_{c,i}$ associated with the i-th axes of the $n+1$ dimensional space in which $\Delta_n^{(\rho)}$ is contained has the following coordinates:

$$[\boldsymbol{p}_{c,i}]_j \begin{cases} 1-(n+1)\cdot\rho & \text{if } j = i, \\ \rho & \text{otherwise.} \end{cases}$$

Which gives a hint of why we need to upper bound ρ by $1/(n+1)$. We define the border $\delta_n^{(\rho)}$ of $\Delta_n^{(\rho)}$ as the union of a set of $n+1$ subspaces ("edges") $\boldsymbol{E}^{(i)}$:

$$\delta_n^{(\rho)} = \bigcup_{i\in[n+1]} \boldsymbol{E}^{(i)},$$

for which each edge $\boldsymbol{E}_\rho^{(i)}$ is defined as a convex combination over all the "corners" of $\Delta_n^{(\rho)}$, excluding the $p_{c,i}$ corner:

$$\boldsymbol{E}_\rho^{(i)} = \Big\{ \sum_{j\in[n+1]\setminus i} \eta_j p_{c,j}, \quad \sum_{j\in[n+1]\setminus i} \eta_i = 1\}.$$

For convenience of notation we also write points on the edge $\boldsymbol{E}_\rho^{(i)}$ as a function of a parameter vector $\boldsymbol{\eta} \in \Delta^{n-1}$. In that setting we write a point on the edge as:

$$\boldsymbol{E}_\rho^{(i)}(\boldsymbol{\eta}) = F_E^{(i)}\boldsymbol{\eta} = \sum_{j\in[n+1]\setminus i} \eta_j \boldsymbol{p}_{c,j}.$$

Note that each edge is thus a convex hull on some set of corners $\boldsymbol{P}_c^{(i)} = \boldsymbol{P}_c \setminus \boldsymbol{p}_{c,i}$ of cardinality n (where $\boldsymbol{P}_c$ is the set of all corners). An illustration making this more intuitive can be found in fig 5.

Proposition 3.1. *(H **is** $\sqrt{m}(\log(\rho^{-1})-1)$-**Lipschitz over** $\Delta_n^{(\rho)}$).*

Proof. This proof is quite direct. We just bound the gradient norm by:

$$\begin{aligned} \|\nabla H(p)\|_2 &= \sqrt{[\nabla H(p)]_1^2 + [\nabla H(p)]_2^2 + [\nabla H(p)]_{n+1}^2} \\ &\leq \sqrt{m} \sup_{\mathbf{p}\in\Delta_n,\ i\in[n+1]} |[\nabla H(p)]_{n+1}| \\ &= \sqrt{m}(-\log\rho - 1) \end{aligned}$$

□

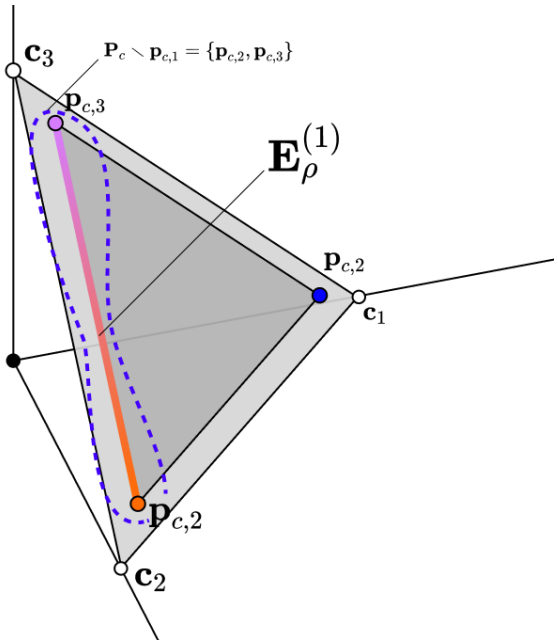

Figure 5: An illustration of how each edge of $\Delta_n^{(\rho)}$ is a convex-hull over a subset of $\Delta_n^{(\rho)}$'s corners.

3.2 Rényi Entropy

An alternative, and more general, in a sense, definition of Entropy is that of the Rényi Entropy. It defines a large class of entropy functions with various properties.

Definition 3.3. *(**Rényi Entropy**) For a discrete random variable X, the Rényi Entropy of order α is defined as:*

$$H_\alpha(X) = \frac{1}{1-\alpha} \log\Big(\sum_{i=1}^{n} p_i^\alpha\Big).$$

Conveniently, it can also be denoted as a function of the L_α norm of the vector of probabilities $\boldsymbol{p}_X$ associated with the random variable X:

$$H_\alpha(X) = H_\alpha(\boldsymbol{p}_X) = \frac{\alpha}{1-\alpha} \log \|\boldsymbol{p}_X\|_\alpha.$$

Observation 3.2. *(**Gradient Properties of the Rényi Entropy of order 2**) The gradient of the Rényi entropy of order 2 (H_2) is simply given by:*

$$\nabla_{\boldsymbol{p}_X} H_2(\boldsymbol{p}_X) = -2\frac{\boldsymbol{p}_X}{\|\boldsymbol{p}_X\|_2^2}.$$

And observing that, on the simplex Δ_n of dimensionality n we have:

$$\begin{aligned}\|\nabla_{\boldsymbol{p}_X} H_2(\boldsymbol{p}_X)\| &= \left\| -2\frac{\boldsymbol{p}_X}{\|\boldsymbol{p}_X\|_2^2} \right\| \\ &= 2\frac{1}{\|\boldsymbol{p}_X\|_2} \leq 2\sqrt{n},\ \ \boldsymbol{p}_X \in \Delta_n,\end{aligned}$$

it is clear that the Rényi entropy is $2\sqrt{n}$-Lipschitz. Furthermore, when computing the eigenvalues the Hessian of H_2, we get:

$$\lambda_{1,2} = \frac{\pm 2}{\|\boldsymbol{p}_X\|_2^2}$$

which, on the Δ_n simplex, is clearly bounded by:

$$\lambda_{1,2} = \frac{\pm 2}{\|\boldsymbol{p}_X\|_2^2} \leq \pm 2n.$$

Which shows H_2 to also be $2n$-smooth.

4 Constrained and Unconstrained Bandits, Iteration Complexity for different Regularizers

In order to work towards a better understanding of the IRL problems at hand we first restrict ourselves to the study of a single-state MDP, which is equivalent to a *K-Armed Bandit Problem*.

4.1 A few observations about Bandit problems

The "*bandit*" (a.k.a single state MDP has a few interesting properties that make the analysis of optimization algorithms on it simpler). In the bandit setting the policy $\boldsymbol{\pi} \in \Delta_A$ becomes simply a distribution on the action-set rather than a function form the states to distributions on the actions-set, we can thus represent $\boldsymbol{\pi}$ as a vector in $\mathbb{R}^n$

Observation 4.1. *(**Policy-Occupancy Measure Equality**) In the specific case of the bandit setting, we have that:*

$$\boldsymbol{\mu}^{\pi} = \boldsymbol{\pi}.$$

This can easily be verified as follows:

$$\begin{aligned}
\mu^{\pi}(a) &= (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} \gamma^t \mathbb{P}_{\nu}^{\pi}(a_t = a) \\
&= (1-\gamma) \lim_{T \to +\infty} \sum_{t=0}^{T} \gamma^t \pi(a_t = a) \\
&= \pi(a).
\end{aligned}$$

Observation 4.2. *(**Objective Function in the Bandit Setting**) In the specific case of the bandit setting, we have that the objective function reduces to:*

$$\begin{aligned}
J^{\nu}(\pi) &= \langle \boldsymbol{r}, \boldsymbol{\mu}^{\pi} \rangle - \tilde{\Omega}(\boldsymbol{\pi}) \\
&= \langle \boldsymbol{r}, \boldsymbol{\pi} \rangle - \Omega(\boldsymbol{\pi}).
\end{aligned}$$

Observation 4.3. *In the bandit setting, the Bellman Flow constraints are always trivially satisfied (the probability that the agent ends up in state s at time $t+1$ is always 1).*

Acceptable policies are defined by the feasible set (as for CRL), which is here given by:

$$\mathcal{F} := \left\{ \boldsymbol{\pi} \in \Delta_A, \Psi \boldsymbol{\pi} \leq \boldsymbol{b} \right\},$$

where Ψ is the cost matrix and $\boldsymbol{b}$ the constraint vector (see section 1.2 for details).

Definition 4.1. *(**Constrained Bandit Solution Map**) we define the constrained bandit solution map $CB : \mathcal{R} \to \Pi$ as:*

$$CB(\boldsymbol{r}) = \arg\max_{\pi \in \mathcal{F}} J(\boldsymbol{\pi}, \boldsymbol{r})$$

(note that here $\mathcal{F} \subseteq \Pi = \mathcal{M} = \Delta_A$).

4.2 Constrained Inverse bandit (CIB)

Definition 4.2. *(**CIRL Solution Map**) The **goal of CIB** is to find some mapping* $CIRL_\pi : \Pi \to \mathcal{R}$.

$$(CB_\pi \circ CIB_\pi)(\boldsymbol{\pi}^E) = \boldsymbol{\pi}^E.$$

We consider the following constraint optimization problem:

$$\min_{\boldsymbol{r}\in\mathcal{R}} \max_{\boldsymbol{\pi}\in\mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\pi} - \boldsymbol{\pi}^E \rangle - \Omega(\boldsymbol{\pi}), \tag{7}$$

which provides a way of evaluating 4.2 (this can be shown to be true by proposition 1.1). The Lagrangian dual of 7, found by relaxing the feasibility constraints is given by:

$$\min_{\boldsymbol{r}\in\mathcal{R},\boldsymbol{\zeta}\geq 0} \max_{\boldsymbol{\pi}\in\Delta_A} f_{\text{CIB}}(\boldsymbol{r}, \boldsymbol{\zeta}, \boldsymbol{\pi}). \tag{8}$$

where $f_{\text{CIB}}(\boldsymbol{r}, \boldsymbol{\zeta}, \boldsymbol{\pi}) = \langle \boldsymbol{r}, \boldsymbol{\pi} - \boldsymbol{\pi}^E \rangle - \Omega(\boldsymbol{\pi}) + \langle \boldsymbol{\zeta}, \boldsymbol{b} - \Psi\boldsymbol{\pi} \rangle$ is our objective function.

Proposition 4.1. *(**Strong Duality**) Assuming that* $\exists \boldsymbol{\pi}^E \in \mathcal{F}$*, and that obj is strictly convex. Then dual optimum is attained for some* $\boldsymbol{\zeta}^* \geq 0$ *and problem 7 is equivalent to an unconstrained bandit problem with reward* $\boldsymbol{r} - \Psi\boldsymbol{\zeta}^*$*. I.e.*

$$CRL_\pi(\boldsymbol{r}) = RL_\pi(\boldsymbol{r} - \boldsymbol{\Psi}\boldsymbol{\zeta}^*).$$

Proof. Using generic Lagrangian Duality theory, we make two observations:

1. observe that the problem: $\max_{\boldsymbol{\pi}\in\mathcal{F}} \langle \boldsymbol{r}, \boldsymbol{\pi} \rangle - \Omega(\boldsymbol{\pi})$, is a **strictly convex optimization problem**,
2. the primal optimum $\boldsymbol{\pi}^*$ is finite as the feasible set $\mathcal{F}$ is bounded and the objective is upper-bounded (since Ω is strictly convex).

From Slater's condition it follows that strong duality holds. □

From proposition 4.1 we know that solving problem 8 is equivalent to solving 7. We will study convergence of extra-gradient-based saddle-point algorithms on 8.

4.2.1 Solving the Shannon Entropy regularized problem with an extra-gradient approach: EG-COP

The most common regularizer used in practice for IRL is the Shannon entropy (see definition 3.1), the use of this particular regularizer poses some problems as it does not have Lipschitz gradients on the domain Δ_A (see discussion in section 3.1). Therefore if we want to show convergence we have to use some trick to ensure gradients are indeed Lipschitz. To go around this limitation we suggest to project the gradients in the ρ-non-vanishing simplex $\Delta_A^{(\rho)} \subset \Delta_A$, on which we can show that our objective function is indeed Lipschitz. In order to more specifically define our problem let us pick a reward class $\mathcal{R}$: we choose our reward class $\mathcal{R}_{L_1}$ to be the L_1 ball centered on the origin, as it closely matches the analysis of identifiability made in [6]. Note that reward class can be easily substituted for another convex reward class without

changing the analysis very much.

We thus consider the following problem:

$$\min_{\boldsymbol{r}\in\mathcal{R},\boldsymbol{\zeta}\geq 0}\max_{\boldsymbol{\pi}\in\Delta_A^{(\rho)}} f_{\text{CIB-H}}(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi}). \tag{9}$$

where $f_{\text{CIB-H}}(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi}) = \langle \boldsymbol{r}, \boldsymbol{\pi}-\boldsymbol{\pi}^E\rangle - H(\boldsymbol{\pi}) + \langle \boldsymbol{\zeta}, \boldsymbol{b}-\Psi\boldsymbol{\pi}\rangle$ is our objective function (the dual Lagrangian of the problem 7). For our result to hold, we need the following assumption:

Assumption 4.1. ***(Sufficient Slater's Condition)*** *we assume that* $\exists\chi>0$ *and* $\pi=\Delta_A$ *s.t.*

$$\boldsymbol{b}-\Psi^T\boldsymbol{\pi}\geq\chi.$$

In order to get a convergence result we need to show that, under our assumptions:

1. The decision variables exist on a domain with a finite bounded diameter (specifically we need to show this for the Lagrange multipliers $\boldsymbol{\zeta}$ which are apparently unbounded in the original problem statement).
2. $\pi^*\in\Delta_A^{(\rho)}$, i.e. the optimal policy lies in our restricted policy set,
3. $f_{\text{CIB-H}}$ is smooth over the domain $\Delta_A^{(\rho)}$,

Proposition 4.2. *Our Lagrange multiplier vector is contained in a box:* $\boldsymbol{\zeta}\in\mathcal{B}$*, where*

$$\mathcal{B}:=\left\{\boldsymbol{\zeta}\in\mathbb{R}^d : 0\leq[\boldsymbol{\zeta}]_i\leq\frac{2(R+\beta\log m)}{\chi(1-\gamma)}\forall i\right\}.$$

Proof. Let $\boldsymbol{Z}_a(\boldsymbol{r}) := \{\boldsymbol{\zeta}\geq 0 : \max_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi})\leq a\}$ be the sublevel set of the dual function $\max_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi})$ for any $a\in\mathbb{R}$, the for any $\boldsymbol{\zeta}\in\boldsymbol{Z}_a$, $\boldsymbol{r}\in\mathcal{R}$ we have:

$$a\geq\max_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r},\boldsymbol{\zeta},\bar{\boldsymbol{\pi}})\geq J(\bar{\boldsymbol{\pi}},\boldsymbol{r})+\zeta^T(b-\Psi^T)\geq\langle\bar{\boldsymbol{\pi}},\boldsymbol{r}\rangle+\chi\zeta^T\mathbf{1}.$$

From which we deduce $[\boldsymbol{\zeta}]_i\leq\frac{a-J(\bar{\boldsymbol{\pi}},\boldsymbol{r})}{\chi}$, choosing $a=J(\boldsymbol{\pi}^*,\boldsymbol{r}^*)$ and using that $J(\boldsymbol{\pi},\boldsymbol{r})\geq\frac{-R-\beta\log m}{1-\gamma}$ we get our result for the upper bound. For the lower bound we just use that $\boldsymbol{\zeta}\geq 0$. □

Proposition 4.3. *The optimal policy* $\boldsymbol{\pi}^*$ *lies in the* ρ*-non-vanishing simplex* $\Delta_A^{(\rho)}$ *with parameter* $\rho=\frac{1}{m}\exp\left(\frac{-2(R+\beta\log(m))(2\|\Psi\|-\chi\gamma+\chi)}{\chi(1-\gamma)^2}\right)$.

Proof. Recall that the optimal policy (for the unconstrained MDP) is given in terms of the optimal Q-values Q^* by :

$$\pi^*(a)=\frac{\exp(Q^*(a)/\beta)}{\sum_{a'\in A}\exp(Q^*(a')/\beta)}. \tag{A}$$

Using from proposition 4.1 that $\mathrm{CRL}_\pi(\boldsymbol{r}) = \mathrm{RL}_\pi(\boldsymbol{r} - \boldsymbol{\Psi}\boldsymbol{\zeta}^*)$ we will use that result (on the modified reward function $\tilde{r}$) to compute our bound. We will start by bounding Q^*:

$$\begin{aligned} Q^*(a) &= \tilde{r}(a) + \gamma \mathbb{E}_{s\sim s_0}\Big[\tilde{V}^*(s)\Big] = r(a) + [\boldsymbol{\Psi}\boldsymbol{\zeta}^*]_a + \tilde{V}^*(s) \\ &= r(a) + [\boldsymbol{\Psi}\boldsymbol{\zeta}^*]_a + \gamma \max_{\boldsymbol{\pi}\in\Delta_A} \mathbb{E}_{s\sim s_0}\Big[\sum_{t=0}^{+\infty} \gamma^t (r(a) + [\boldsymbol{\Psi}\boldsymbol{\zeta}^*]_a + \beta H(\boldsymbol{\pi}))\Big]. \end{aligned}$$

We now let $\alpha = \frac{R+\beta\log m}{1-\gamma}$ and claim that $-\alpha \le Q(a) \le \alpha$, this can be verified by:

$$\begin{aligned} Q^*(a) &\le \max_{\boldsymbol{\pi}\in\Delta_A} \left[\sum_{t=0}^{+\infty} \gamma^t \big(r(a) + [\boldsymbol{\Psi}\boldsymbol{\zeta}^*]_a + \beta H(\boldsymbol{\pi})\big)\right] \\ &\le \Big(\sum_{t=0}^{+\infty} \gamma^t\Big)\Big(R + \|\Psi\| \sup_{\boldsymbol{\zeta}^*} \|\boldsymbol{\zeta}^*\|_\infty + \beta\log m\Big) \\ &= \frac{(R+\beta\log(m))\,(2\|\Psi\| - \chi\gamma + \chi)}{\chi\,(1-\gamma)^2} = \alpha, \\ Q^*(a) &\ge \min_{\boldsymbol{\pi}\in\Delta_A} \left[\sum_{t=0}^{+\infty} \gamma^t \big(r(a) + [\boldsymbol{\Psi}\boldsymbol{\zeta}^*]_a + \beta H(\boldsymbol{\pi})\big)\right] \\ &\ge \Big(\sum_{t=0}^{+\infty} \gamma^t\Big)\Big(-R\Big) \\ &\ge \Big(\sum_{t=0}^{+\infty} \gamma^t\Big)\Big(-R - \beta\log m\Big) \ge -\alpha. \end{aligned}$$

Thus using (A) we have that:

$$\pi^*(a) \ge \frac{\exp(-\alpha)}{\sum'_a \exp(\alpha)} = \frac{\exp -2\alpha}{m} = \frac{1}{m}\exp\left(\frac{-2\,(R+\beta\log(m))\,(2\|\Psi\| - \chi\gamma + \chi)}{\chi\,(1-\gamma)^2}\right).$$

Which completes the proof (by definition of the the ρ-non-vanishing simplex). □

Proposition 4.4. *$f_{CIB\text{-}H}$ is smooth with constant $L = m\exp\left(\frac{2(R+\beta\log(m))(2\|\Psi\|-\chi\gamma+\chi)}{\chi(1-\gamma)^2}\right)$.*

Proof. We look at $\nabla^2 f_{\text{CIB-H}}$ the hessian of the objective function. Direct computation shows that it is a diagonal matrix of the form :

$$[\nabla^2 f(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi})]_{i,j} = \begin{cases} \frac{1}{p_i} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

The diagonal form of $\nabla^2 f(\boldsymbol{r},\boldsymbol{\zeta},\boldsymbol{\pi})$ makes it trivial to bound it's spectral norm (the spectral norm is whatever element of the diagonal is maximize). It is thus just a matter of bounding $\frac{1}{p_i}$ which we can do using proposition 4.3:

$$\|\nabla^2 f\| \le \max \mathrm{eig}\Big(\nabla^2 f\Big) \le m\exp\left(\frac{2\,(R+\beta\log(m))\,(2\|\Psi\| - \chi\gamma + \chi)}{\chi\,(1-\gamma)^2}\right).$$

□

Now we tackle showing that the diameter is bounded, since our reward class is a ball and since our policies are constrained to the ρ-non-vanishing simplex, this is only a matter of showing that the Lagrange multipliers $\boldsymbol{\zeta}$ are bounded to some box. Which is what we do in the next proposition.

At this point we are ready to describe the algorithm that we will use to solve the CIRL problem. It is simply a specific application of projected extra-gradient GDA (see alg. 2 in section 2.3) to the objective function $f_{\text{CIB-H}}$ where we project on the domain $\mathcal{R}_{L_1} \times \mathcal{B} \times \Delta_A^{(\rho)}$. Theorem 4.1 proves that:

1. the algorithm converges in finite time, with a $O(1/\epsilon)$ rate,
2. the algorithm recovers the exact solution (up to an error ϵ) if the problem is identifiable.

Algorithm 4: EG-COP: Extra-gradient constrained inverse bandit algorithm

Set the learning rate $\eta > 0$
Initialize the algorithm at some point $(\boldsymbol{r}_0, \boldsymbol{\zeta}_0, \boldsymbol{\pi}_0)$
foreach *iteration* $k = 0, 2, ..., K-1$ **do**
$\boldsymbol{r}_{k+1/2} \leftarrow \Pi_{\mathcal{R}_{L_1}}\big(\boldsymbol{r}_k - \eta \nabla_{\boldsymbol{r}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\big)$
$\boldsymbol{\zeta}_{k+1/2} \leftarrow \Pi_{\boldsymbol{\zeta} \in \mathcal{B}}\big(\boldsymbol{\zeta}_k - \eta \nabla_{\boldsymbol{\zeta}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\big)$
$\boldsymbol{\pi}_{k+1/2} \leftarrow \Pi_{\Delta_A^{(\rho)}}\big(\boldsymbol{\pi}_k + \eta \nabla_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\big)$
$\boldsymbol{r}_{k+1} \leftarrow \Pi_{\mathcal{R}_{L_1}}\big(\boldsymbol{r}_k - \eta \nabla_{\boldsymbol{r}} f_{\text{CIB-H}}(\boldsymbol{r}_{k+1/2}, \boldsymbol{\zeta}_{k+1/2}, \boldsymbol{\pi}_{k+1/2})\big)$
$\boldsymbol{\zeta}_{k+1} \leftarrow \Pi_{\boldsymbol{\zeta} \in \mathcal{B}}\big(\boldsymbol{\zeta}_k - \eta \nabla_{\boldsymbol{\zeta}} f_{\text{CIB-H}}(\boldsymbol{r}_{k+1/2}, \boldsymbol{\zeta}_{k+1/2}, \boldsymbol{\pi}_{k+1/2})\big)$
$\boldsymbol{\pi}_{k+1} \leftarrow \Pi_{\Delta_A^{(\rho)}}\big(\boldsymbol{\pi}_k + \eta \nabla_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r}_{k+1/2}, \boldsymbol{\zeta}_{k+1/2}, \boldsymbol{\pi}_{k+1/2})\big)$
end
return $(\hat{\boldsymbol{r}}_K, \hat{\boldsymbol{\zeta}}_K, \hat{\boldsymbol{\pi}}_K)$, where $\hat{\cdot}$ denotes the empirical mean over a sequence.

Theorem 4.1. *Algorithm 4 recovers the optimal reward and policy (up to reward shaping transformations), up to approximation error ϵ in time $O(1/K)$ (where K is the number of iterations). More specifically, assuming we choose learning rate $\eta = \frac{1}{2m} \exp\left(\frac{-2(R+\beta \log(m))(2\|\Psi\| - \chi\gamma + \chi)}{\chi(1-\gamma)^2}\right)$, we have:*

$$DG(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k) \leq C_{\textit{EG-COP}} \frac{1}{K},$$

where:

$$C_{\textit{EG-COP}} = 2mD^2 \exp\left(\frac{2\left(R + \beta \log(m)\right)\left(2\|\Psi\| - \chi\gamma + \chi\right)}{\chi\left(1-\gamma\right)^2}\right).$$

In which:

$$D = \max\left\{\sqrt{m}, 2R + \frac{2(R + \beta \log m)}{\chi(1-\gamma)}\right\}.$$

Proof. The proof follows the analysis of alg. 2, which we do not re-state for brevity, so we just have to verify that the assumptions are satisfied:

1. we note that the function $f_{\text{CIB-H}}$ is concave-convex in $(\boldsymbol{r}, \boldsymbol{\zeta}, \boldsymbol{\pi})$ (where we just stack $\boldsymbol{r}$ and $\boldsymbol{\zeta}$ to have a saddle-point formulation),

2. we have that the domain on which the variables are defined is bounded by $D = \max\left\{\sqrt{m}, 2R + \frac{2(R+\beta\log m)}{\chi(1-\gamma)}\right\}$ (from the definitions of the reward class $\mathcal{R}_{L_1}$, the fact that the policy in in the non-vanishing simplex is contained in the simplex and thus has diameter $< \sqrt{m}$ and using the result of proposition 4.1 to get the diameter of $\mathcal{B}$),

3. we known that the function $f_{\text{CIB-H}}$ is smooth with parameter

$$m \exp\left(\frac{2\left(R + \beta\log\left(m\right)\right)\left(2\|\Psi\| - \chi\gamma + \chi\right)}{\chi\left(1-\gamma\right)^2}\right)$$

   from proposition 4.4.

Plugging those values into proposition 2.3 gives us our convergence time. To verify that what is recovered is indeed correct we refer to proposition 1.1 (which applies as the bandit case is simply a restriction of the generic IRL case). The choice of learning rate simply comes from the result of proposition 4.4 together with the required assumptions of proposition 1.1. □

4.2.2 Solving the Shannon-regularized problem with gradient descent-ascent : GDA-COP

The very large constant in front of the EG-COP algorithm's rate motivates the search for alternative methods. The most straight-forward approach to that problem is to simply take average iterates of the GDA algorithm (using the same projections as in algorithm 4).

Algorithm 5: GDA-COP: Gradient-Descent Ascent constrained inverse bandit algorithm

Set the learning rate $\eta_k > 0$
Initialize the algorithm at some point $(\boldsymbol{r}_0, \boldsymbol{\zeta}_0, \boldsymbol{\pi}_0)$
foreach *iteration* $k = 0, 2, ..., K-1$ **do**
  $\eta_k \to \xi\frac{1}{\sqrt{k}}$
  $\boldsymbol{r}_{k+1} \leftarrow \Pi_{\mathcal{R}_{L_1}}\left(\boldsymbol{r}_k - \eta_k \nabla_{\boldsymbol{r}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\right)$
  $\boldsymbol{\zeta}_{k+1} \leftarrow \Pi_{\boldsymbol{\zeta}\in\mathcal{B}}\left(\boldsymbol{\zeta}_k - \eta_k \nabla_{\boldsymbol{\zeta}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\right)$
  $\boldsymbol{\pi}_{k+1} \leftarrow \Pi_{\Delta_A^{(\rho)}}\left(\boldsymbol{\pi}_k + \eta_k \nabla_{\boldsymbol{\pi}} f_{\text{CIB-H}}(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k)\right)$
end
return $(\hat{\boldsymbol{r}}_K, \hat{\boldsymbol{\zeta}}_K, \hat{\boldsymbol{\pi}}_K)$, where $\hat{\cdot}$ denotes the empirical mean over a sequence.

Theorem 4.2. *Algorithm 5 recovers the optimal reward and policy (up to reward shaping transformations), up to approximation error ϵ in time $O(1/\sqrt{K})$ (where K is the number of iterations). More specifically, assuming we use the decreasing learning rate $\eta_k = \xi\frac{1}{\sqrt{k}}$, we have:*

$$DG(\boldsymbol{r}_k, \boldsymbol{\zeta}_k, \boldsymbol{\pi}_k) \leq C_{GDA\text{-}COP}\frac{1}{\sqrt{K}},$$

where:

$$C_{GDA\text{-}COP} = 4\sqrt{m}D\Big(\log m + \frac{2\left(R + \beta\log\left(m\right)\right)\left(2\|\Psi\| - \chi\gamma + \chi\right)}{\chi\left(1-\gamma\right)^2} - 1\Big).$$

In which:

$$D = \max\Big\{\sqrt{m}, 2R + \frac{2(R + \beta\log m)}{\chi(1-\gamma)}\Big\},$$

is the domain diameter and:

$$\xi = \frac{D}{\sqrt{2m}\Big(\log m + \frac{2(R+\beta\log\left(m\right))(2\|\Psi\|-\chi\gamma+\chi)}{\chi(1-\gamma)^2} - 1\Big)}.$$

Proof. The proof follows the analysis of alg. 1, which we do not re-state for brevity, so we just have to verify that the assumptions are satisfied:

1. we note that the function $f_{\text{CIB-H}}$ is concave-convex in $(\boldsymbol{r}, \boldsymbol{\zeta}, \boldsymbol{\pi})$ (where we just stack $\boldsymbol{r}$ and $\boldsymbol{\zeta}$ to have a saddle-point formulation),

2. we have that the domain on which the variables are defined is bounded by $D = \max\Big\{\sqrt{m}, 2R + \frac{2(R+\beta\log m)}{\chi(1-\gamma)}\Big\}$ (from the definitions of the reward class $\mathcal{R}_{L_1}$, the fact that the policy in in the non-vanishing simplex is contained in the simplex and thus has diameter $< \sqrt{m}$ and using the result of proposition 4.1 to get the diameter of $\mathcal{B}$),

3. we known that the function $f_{\text{CIB-H}}$ is Lipschitz with parameter

   $$\sqrt{m}\Big(\log m + \frac{2\left(R + \beta\log\left(m\right)\right)\left(2\|\Psi\| - \chi\gamma + \chi\right)}{\chi\left(1-\gamma\right)^2} - 1\Big).$$

   from the Lipschitz gradient result of proposition 3.1 and using the ρ bound of proposition 4.3.

Plugging those values into proposition 2.2 gives us our convergence time. To verify that what is recovered is indeed correct we refer to proposition 1.1 (which applies as the bandit case is simply a restriction of the generic IRL case). The choice of learning rate simply comes from the result of proposition 2.2 together with the required assumptions of proposition 1.1. □

References

[1] Maurice Sion. "On general minimax theorems". en. In: *Pacific Journal of Mathematics* 8.1 (Mar. 1958), pp. 171–176. ISSN: 0030-8730, 0030-8730. DOI: 10.2140/pjm.1958.8.171. URL: http://msp.org/pjm/1958/8-1/p14.xhtml (visited on 03/20/2023).

[2] Alekh Agarwal et al. "On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift". In: *Proceedings of Machine Learning Research* (2020).

[3] Mokhtari Aryan, Ozdaglar Asuman E., and Pattathil Sarath. "Convergence Rate of O(1/k) for Optimistic Gradient and Extragradient Methods in Smooth Convex-Concave Saddle Point Problems". In: *SIAM Journal on Optimization* (2020). DOI: 10.1137/19M127375X.

[4] Yang Cai, Argyris Oikonomou, and Weiqiang Zheng. "Tight Last-Iterate Convergence of the Extragradient and the Optimistic Gradient Descent-Ascent Algorithm for Constrained Monotone Variational Inequalities". In: *Advances in Neural Information Processing Systems*. 2022.

[5] Eduard Gorbunov, Adrien Taylor, and Gauthier Gidel. "Last-Iterate Convergence of Optimistic Gradient Method for Monotone Variational Inequalities". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022.

[6] Andreas Schlaginhaufen. "Identifiability and generalizability in constrained inverse reinforcement learning". In: *preprint* (2023).