Side Information in Inverse Optimal Control

Ivan Papusha (UT Austin)
Min Wen (UPenn)
Ufuk Topcu (UT Austin)

Problem description

- Incorporating **expert demonstrations** into an autonomous system is difficult
- Even when expert demonstrations are somehow incorporated, generalizing to unseen scenarios can be **unsafe**.
- Can we generalize in a “safe” way using side information?
  - what does “safe” mean?
  - what kind of “side information” can be incorporated?

Outline

1. Parameterized optimization

2. Learning from expert demonstrations

3. Incorporating side specifications

Parameterized optimization

Forward problem:

- f(x, p) is a convex function of x for every p
- given p, find the optimal x*
- optimality condition:

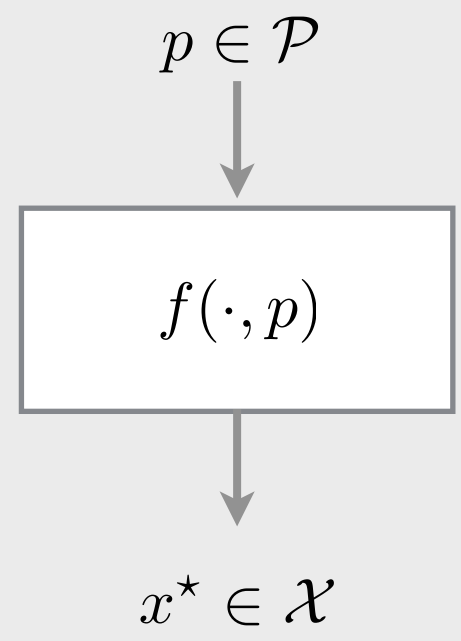
$$\nabla_x f(x^\star, p) = 0$$

Inverse problem:

- given a demonstration data set, determine f
- data set contains **allegedly** optimal points $x^{(k)}$ for every parameter $p^{(k)}$

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \dots, N \right\}$$

Parameterized optimization



Forward problem:

- f(x, p) is a convex function of x for every p
- given p, find the optimal x*
- optimality condition:

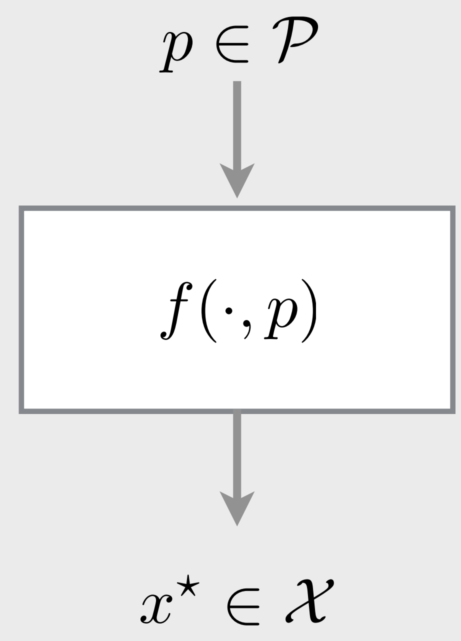
$$\nabla_x f(x^\star, p) = 0$$

Inverse problem:

- given a demonstration data set, determine f
- data set contains **allegedly** optimal points $x^{(k)}$ for every parameter $p^{(k)}$

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

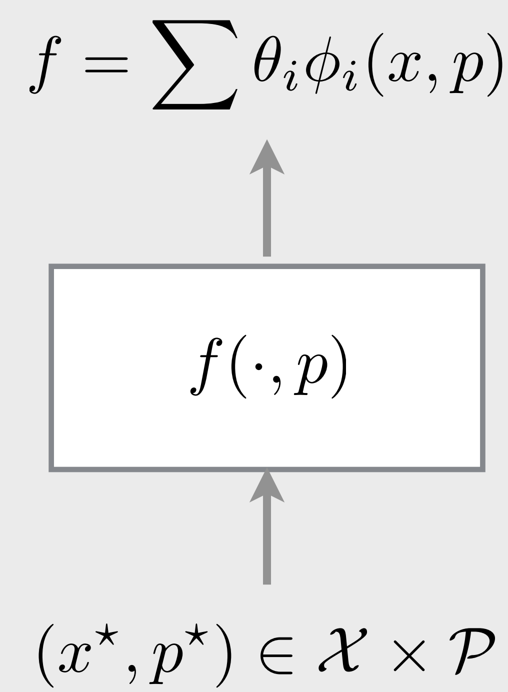
Parameterized optimization



Forward problem:

- $f(x, p)$ is a convex function of x for every p
- given p, find the optimal x^*
- optimality condition:

$$\nabla_x f(x^\star, p) = 0$$



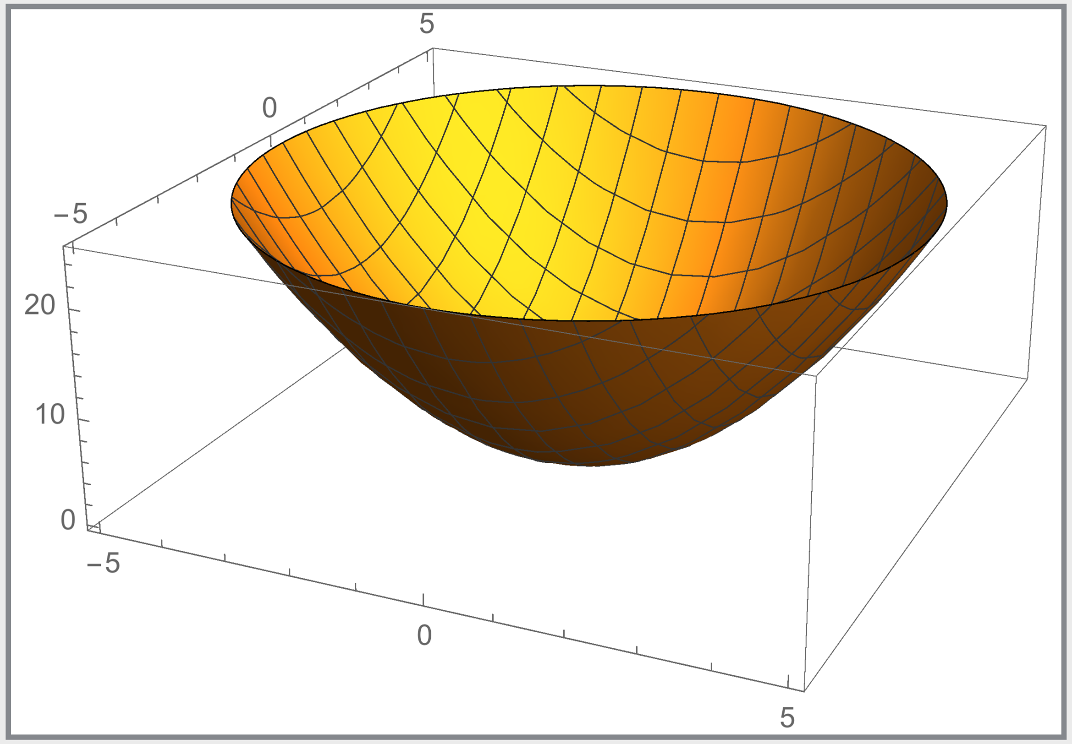
Inverse problem:

- given a demonstration data set, determine f
- data set contains **allegedly** optimal points $x^{(k)}$ for every parameter $p^{(k)}$

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Forward optimization

Objective	**Parameters**	**Optimal value**
$f(x,y) = x^2 + y^2$	$\alpha = 0, \beta = 0$	$x^\star = 0, y^\star = 0$
$f(x,y) = (x-1)^2 + y^2$	$\alpha = 1, \beta = 0$	$x^\star = 1, y^\star = 0$
$f(x,y) = (x-\alpha)^2 + (y-\beta)^2$	α, β	$x^\star = \alpha, y^\star = \beta$



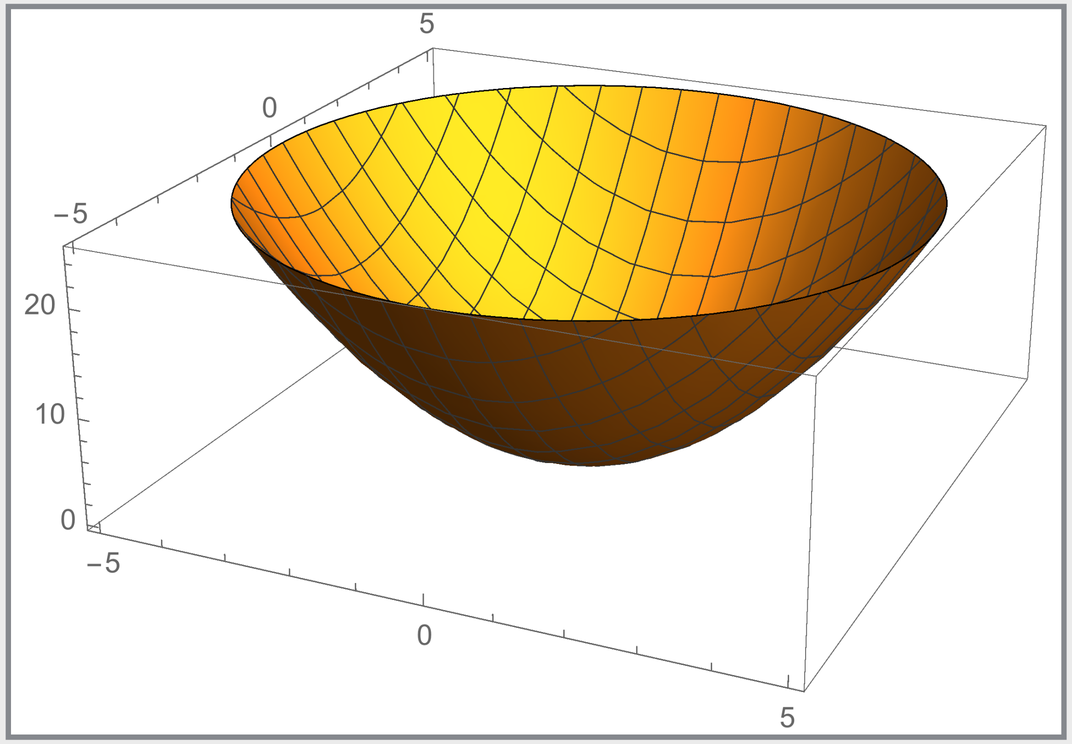
Forward optimization

Objective → **Parameters** → **Optimal value**

$f(x,y) = x^2 + y^2$	$\alpha = 0, \beta = 0$	$x^\star = 0, y^\star = 0$
$f(x,y) = (x-1)^2 + y^2$	$\alpha = 1, \beta = 0$	$x^\star = 1, y^\star = 0$
$f(x,y) = (x-\alpha)^2 + (y-\beta)^2$	α, β	$x^\star = \alpha, y^\star = \beta$

Optimality condition:

$$\nabla_{(x,y)} f(x,y) = \begin{bmatrix} 2x - 2\alpha \\ 2y - 2\beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Forward optimization

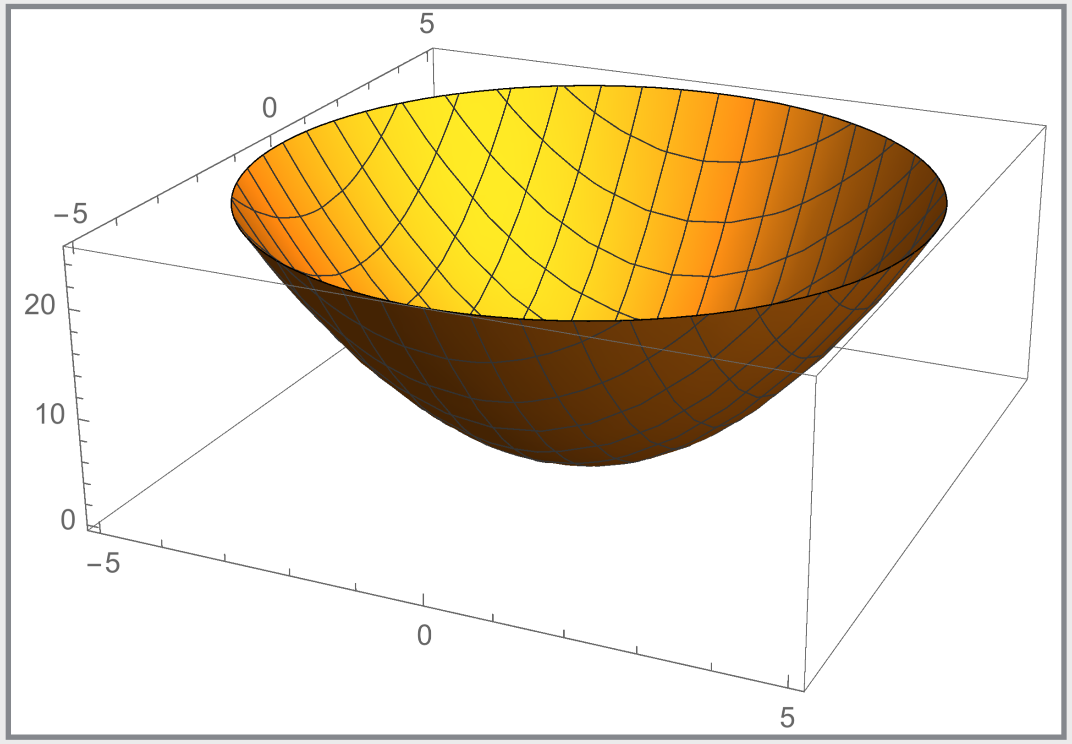
Objective	**Parameters**	**Optimal value**
$f(x,y) = x^2 + y^2$	$\alpha = 0, \beta = 0$	$x^\star = 0, y^\star = 0$
$f(x,y) = (x-1)^2 + y^2$	$\alpha = 1, \beta = 0$	$x^\star = 1, y^\star = 0$
$f(x,y) = (x-\alpha)^2 + (y-\beta)^2$	α, β	$x^\star = \alpha, y^\star = \beta$

Optimality condition:

$$\nabla_{(x,y)} f(x,y) = \begin{bmatrix} 2x - 2\alpha \\ 2y - 2\beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Stationarity residual:

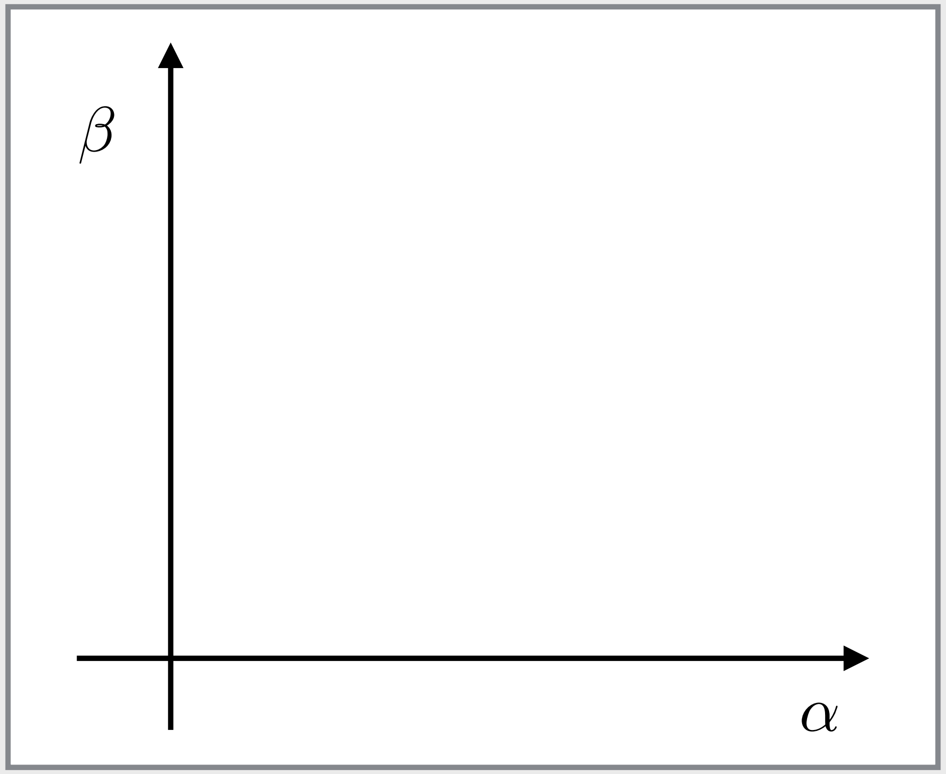
$$r_{\text{stat}}^{(x,y)}(\alpha,\beta) = \begin{bmatrix} 2x - 2\alpha \\ 2y - 2\beta \end{bmatrix}$$



Inverse optimization

Objective ← **Parameters** ← **Optimal value**

$$f(x,y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -\alpha \\ 0 & 1 & -\beta \\ -\alpha & -\beta & \alpha^2+\beta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



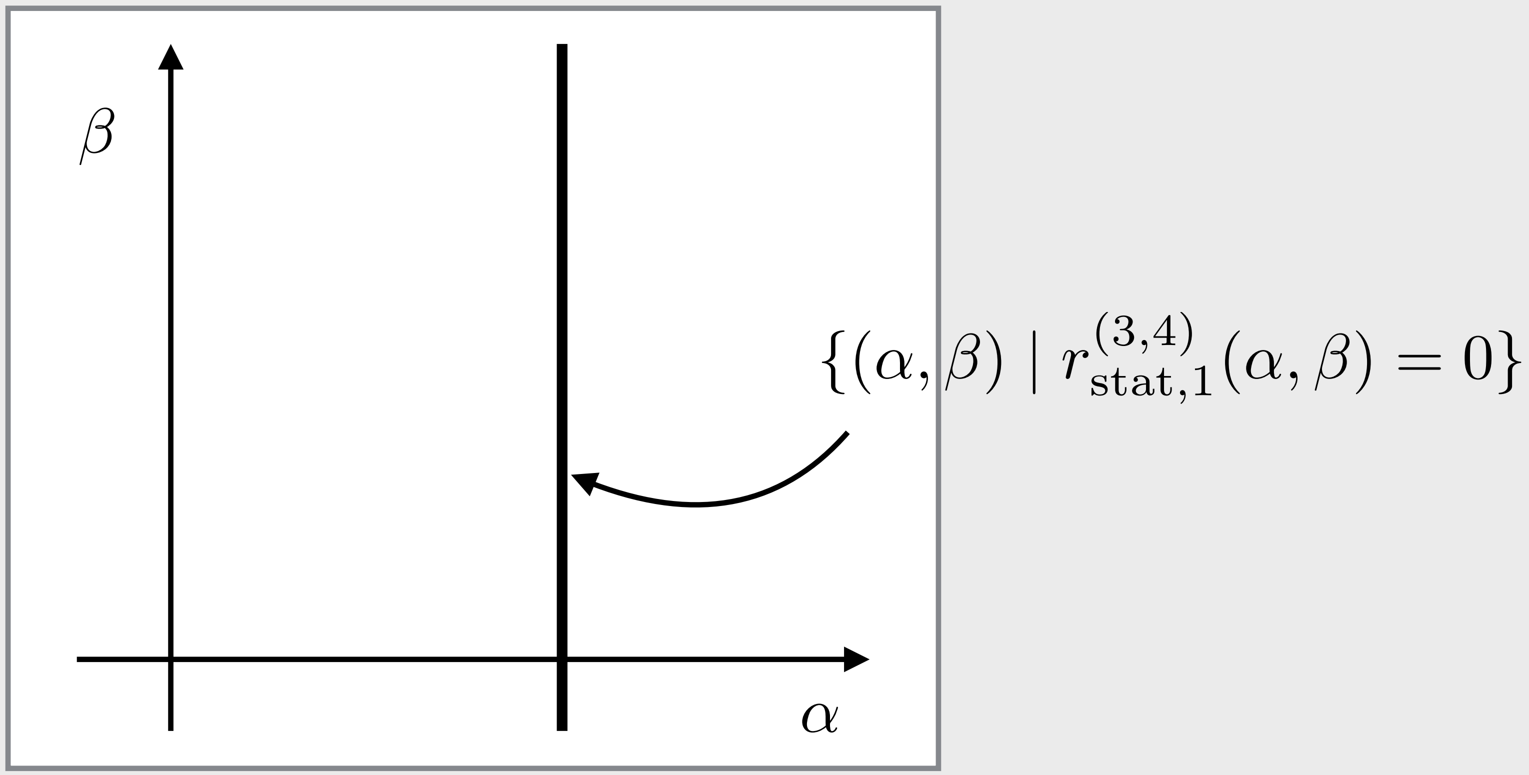
Inverse optimization

Objective ← **Parameters** ← **Optimal value**

$$f(x,y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -\alpha \\ 0 & 1 & -\beta \\ -\alpha & -\beta & \alpha^2+\beta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\alpha = 3$$

$$x^\star = 3$$



Inverse optimization

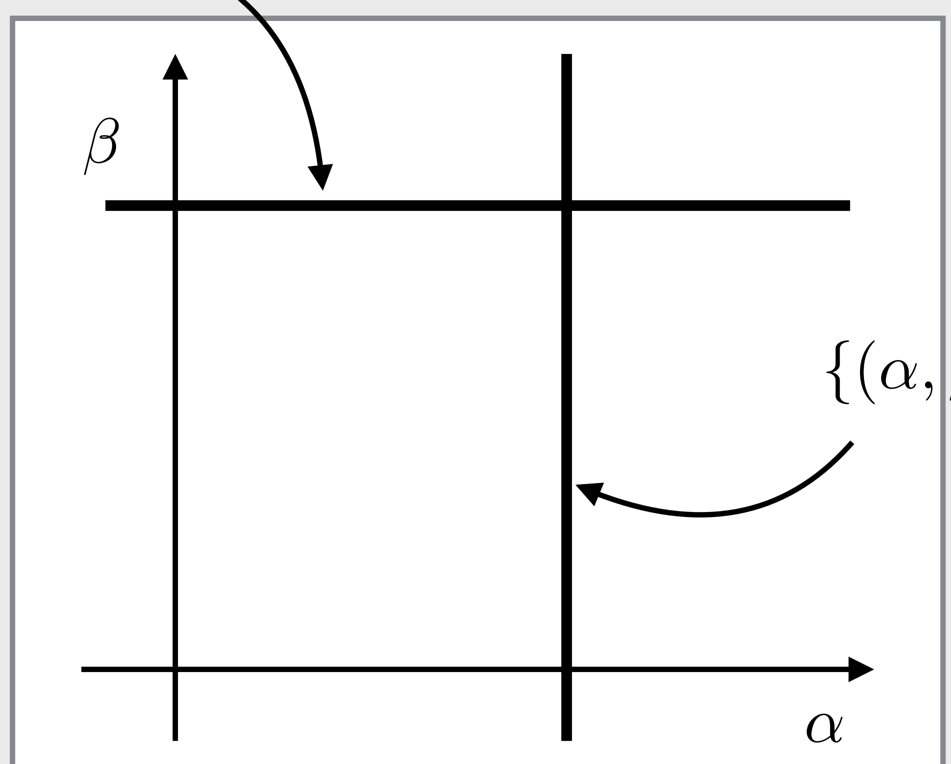
Objective ← **Parameters** ← **Optimal value**

$$f(x,y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -\alpha \\ 0 & 1 & -\beta \\ -\alpha & -\beta & \alpha^2+\beta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\alpha = 3 \qquad \beta = 4$$

$$x^\star = 3 \qquad y^\star = 4$$

$$\{(\alpha,\beta) \mid r^{(3,4)}_{\mathrm{stat},2}(\alpha,\beta) = 0\}$$



$$\{(\alpha,\beta) \mid r^{(3,4)}_{\mathrm{stat},1}(\alpha,\beta) = 0\}$$

Inverse optimization

Objective ← **Parameters** ← **Optimal value**

$$f(x,y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -\alpha \\ 0 & 1 & -\beta \\ -\alpha & -\beta & \alpha^2+\beta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

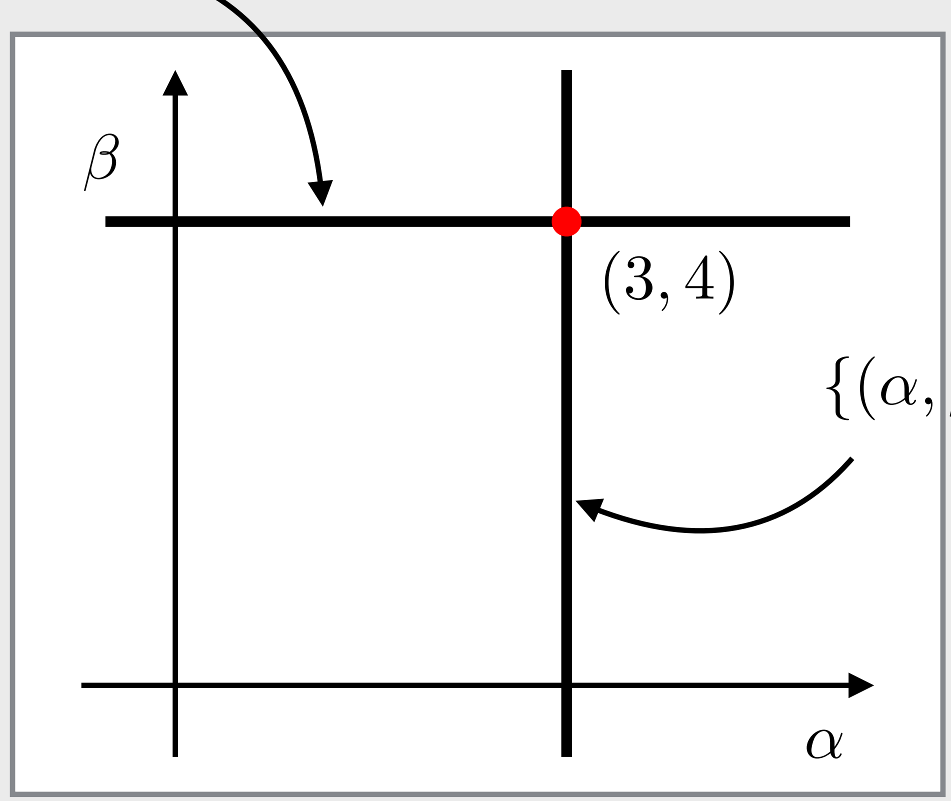
$\alpha = 3$

$\beta = 4$

$x^\star = 3$

$y^\star = 4$

$\{(\alpha,\beta) \mid r^{(3,4)}_{\mathrm{stat},2}(\alpha,\beta) = 0\}$



$\{(\alpha,\beta) \mid r^{(3,4)}_{\mathrm{stat},1}(\alpha,\beta) = 0\}$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p) \qquad\qquad \mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

data set of optimal points and parameters

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p) \qquad \mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

coefficients to determine

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p)$$

data set of optimal points and parameters

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

coefficients to determine

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p)$$

known basis functions, e.g., x^2

data set of optimal points and parameters

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

coefficients to determine

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p)$$

known basis functions, e.g., x^2

data set of optimal points and parameters

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Consistence with optimality:

- the basis coefficients are consistent with the optimality the *k*th data point $(x^{(k)},p^{(k)})$ if the *k*th residual is zero:

$$r_{\text{stat}}^{(k)}(\theta) = \sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$$

Imputing an objective

In general:

- assume f(x,p) is a linear combination of bases
- must determine the basis coefficients **consistent with the optimality** of every point in the data set D

coefficients to determine

$$f(x,p) = \sum_{i=1}^{M} \theta_i \phi_i(x,p)$$

known basis functions, e.g., x^2

data set of optimal points and parameters

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

Consistence with optimality:

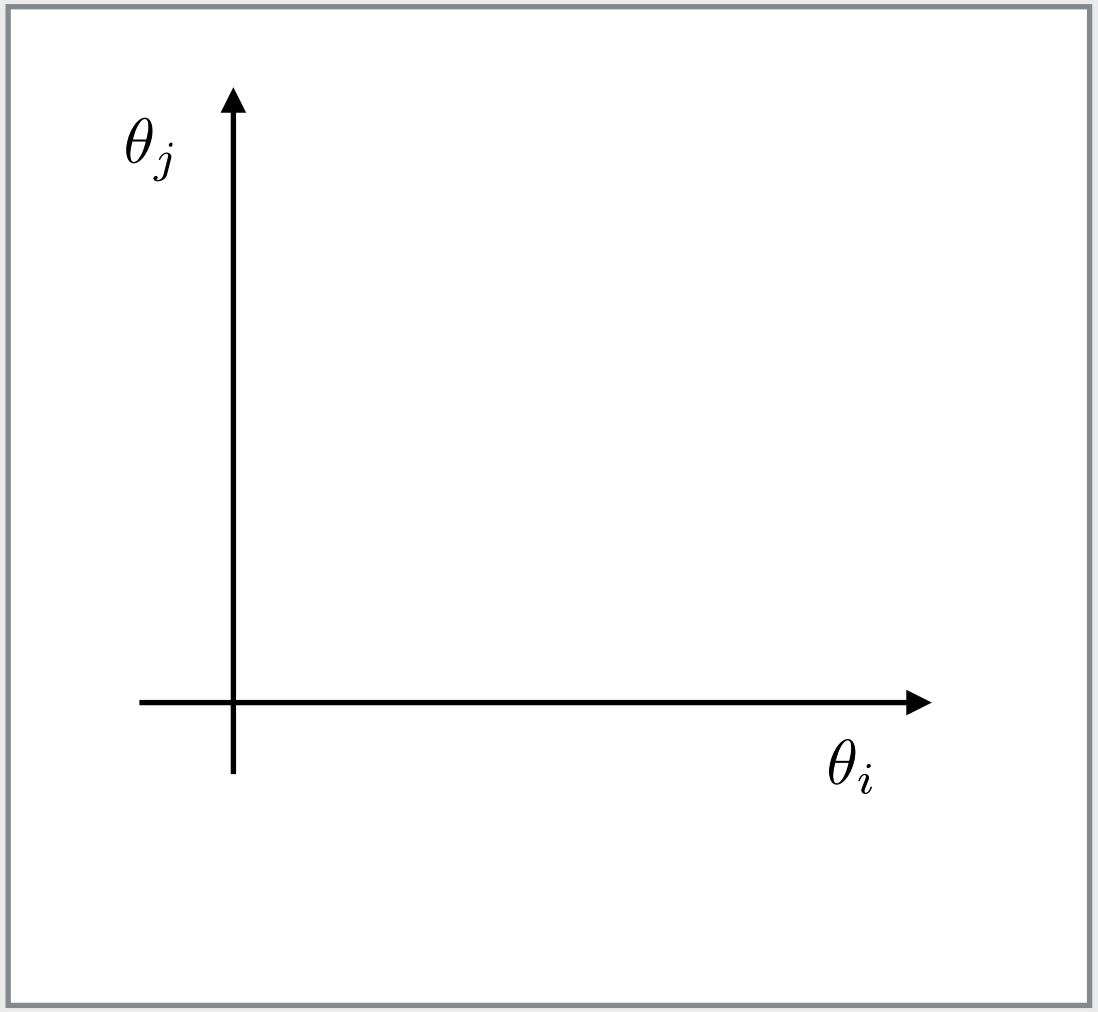
- the basis coefficients are consistent with the optimality the *k*th data point $(x^{(k)}, p^{(k)})$ if the *k*th residual is zero:

$$r_{\text{stat}}^{(k)}(\theta) = \sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$$

linear constraint on θ

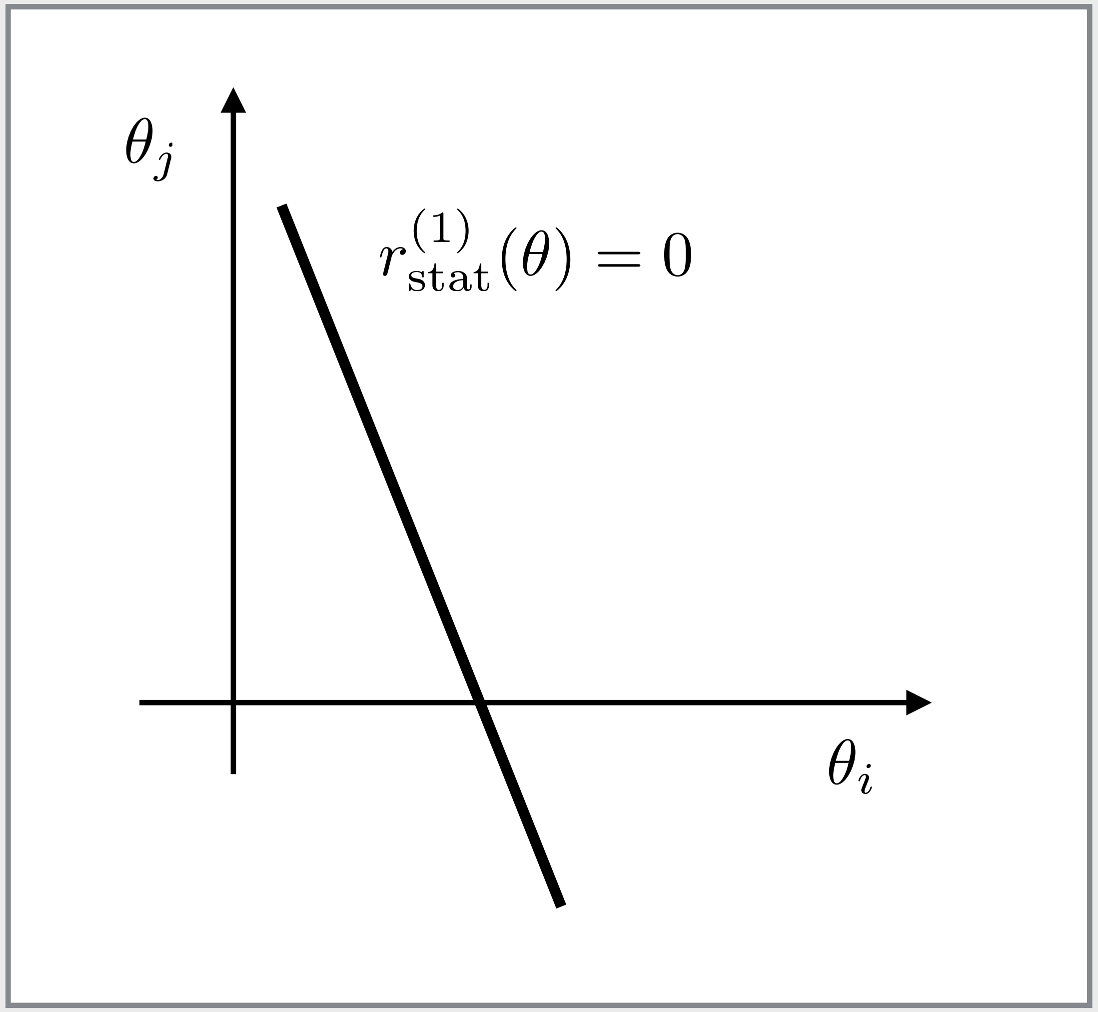
Consistency with every expert example

Data point **Linear constraint**



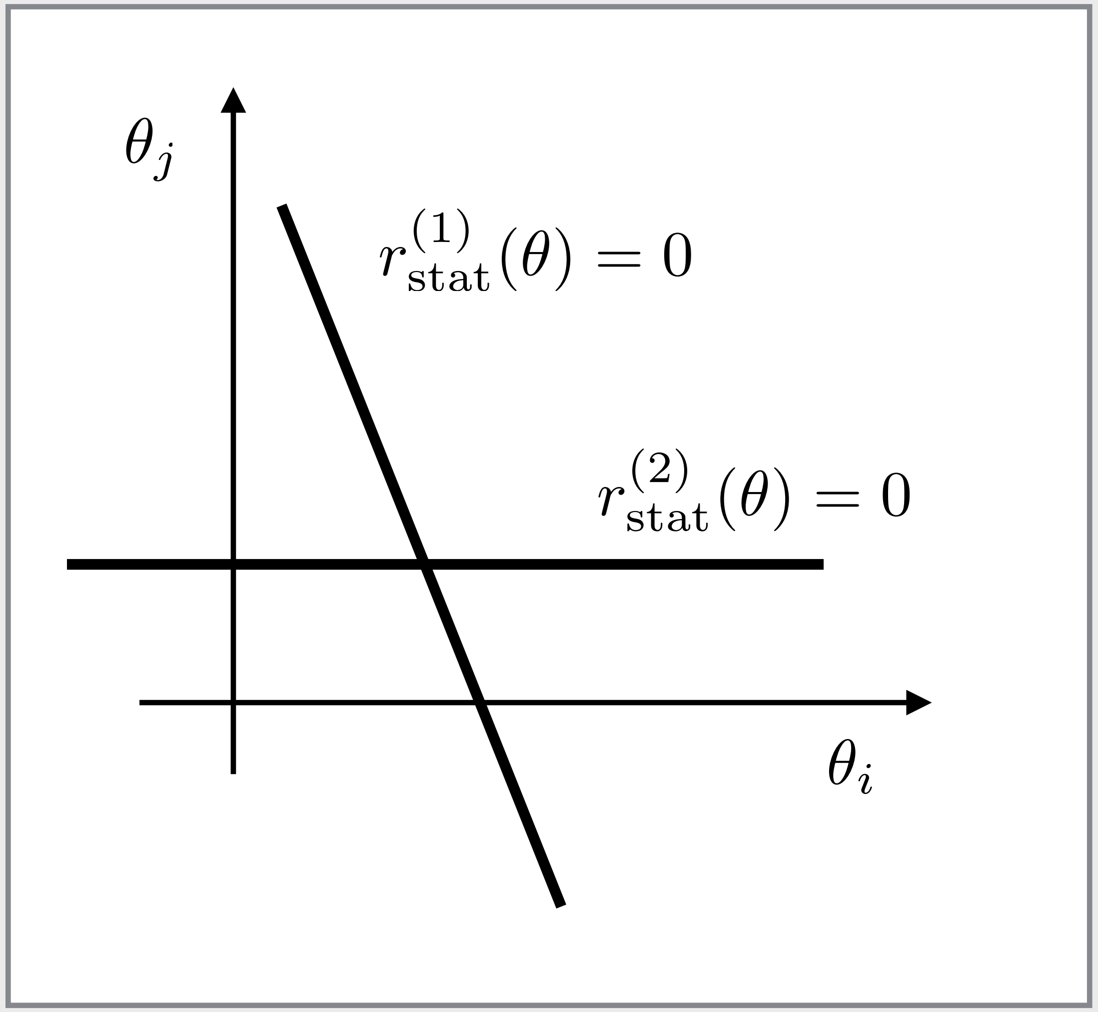
Consistency with every expert example

Data point	**Linear constraint**
$k = 1$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(1)}, p^{(1)}) = 0$



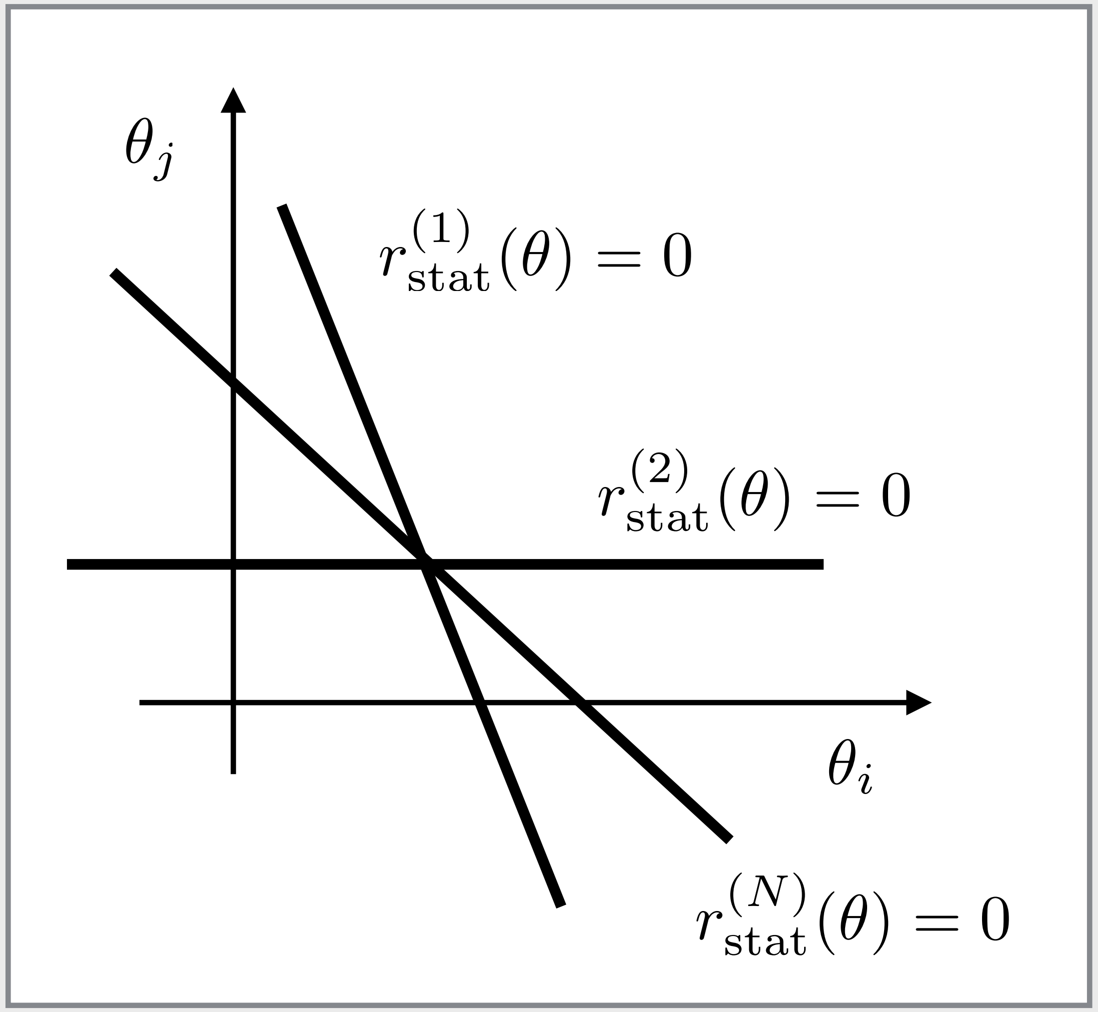
Consistency with every expert example

Data point	Linear constraint
$k = 1$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(1)}, p^{(1)}) = 0$
$k = 2$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(2)}, p^{(2)}) = 0$



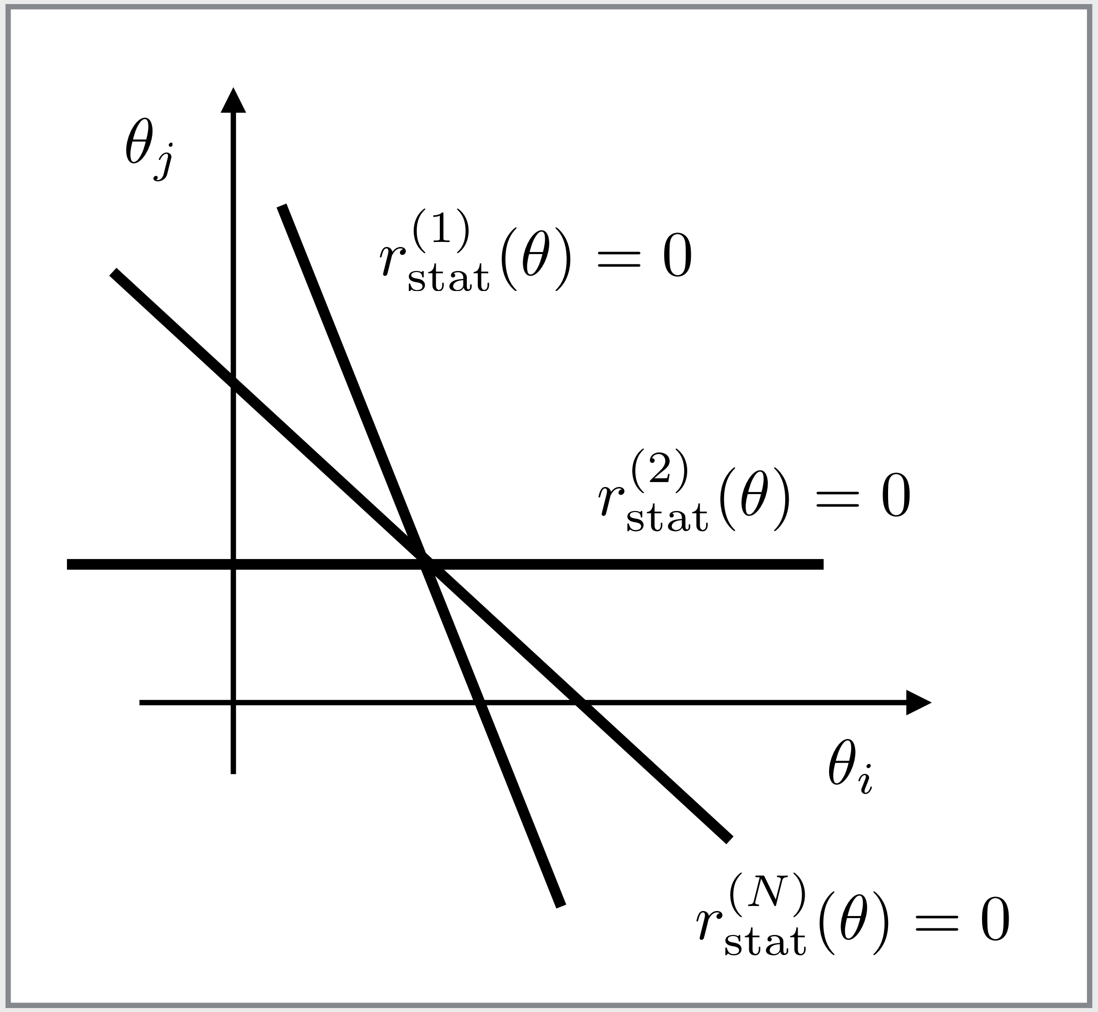
Consistency with every expert example

Data point	Linear constraint
$k = 1$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(1)}, p^{(1)}) = 0$
$k = 2$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(2)}, p^{(2)}) = 0$
$\vdots$	$\vdots$
$k = N$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(N)}, p^{(N)}) = 0$



Consistency with every expert example

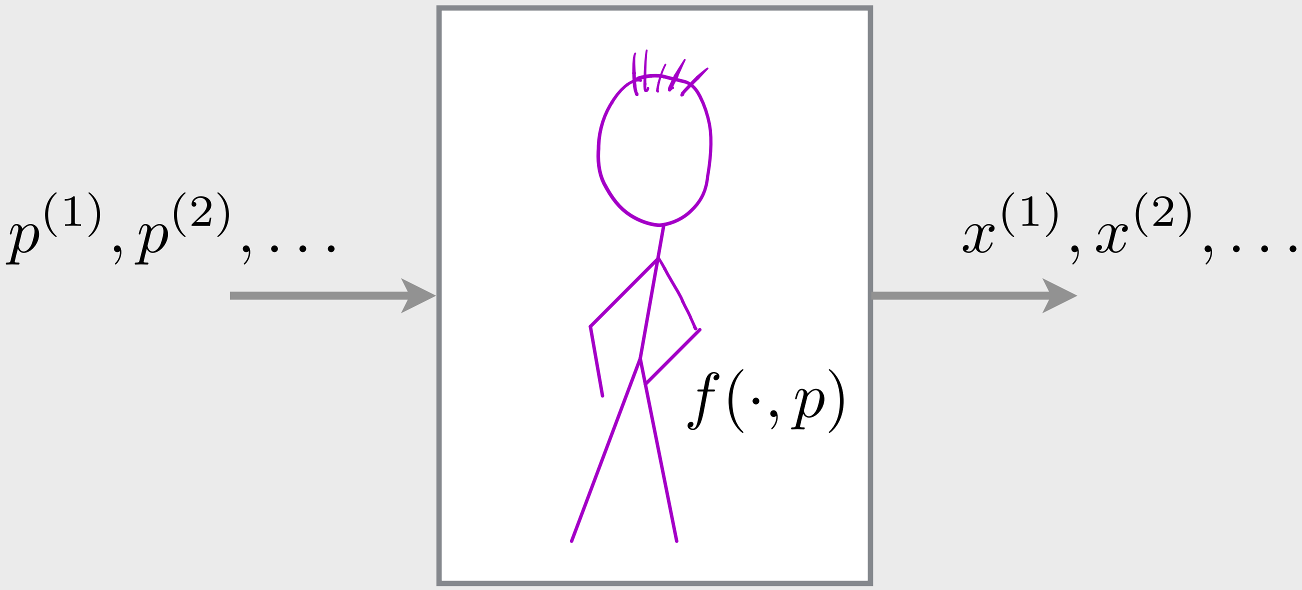
Data point	Linear constraint
$k = 1$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(1)}, p^{(1)}) = 0$
$k = 2$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(2)}, p^{(2)}) = 0$
$\vdots$	$\vdots$
$k = N$	$\sum_{i=1}^{M} \theta_i \nabla_x \phi_i(x^{(N)}, p^{(N)}) = 0$



N*n constraints on M values

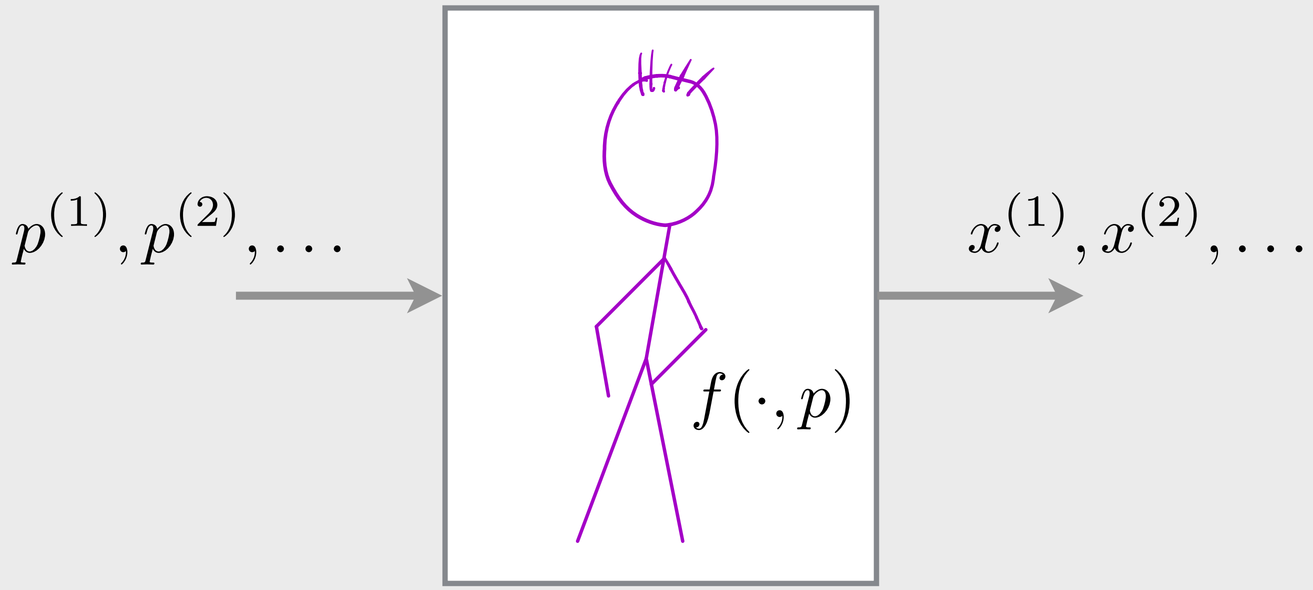
Typical picture

- What if the points $x^{(k)}$ are only **approximately** optimal for every parameter $p^{(k)}$?
- for example, what if the "optimizer" is a human?



Typical picture

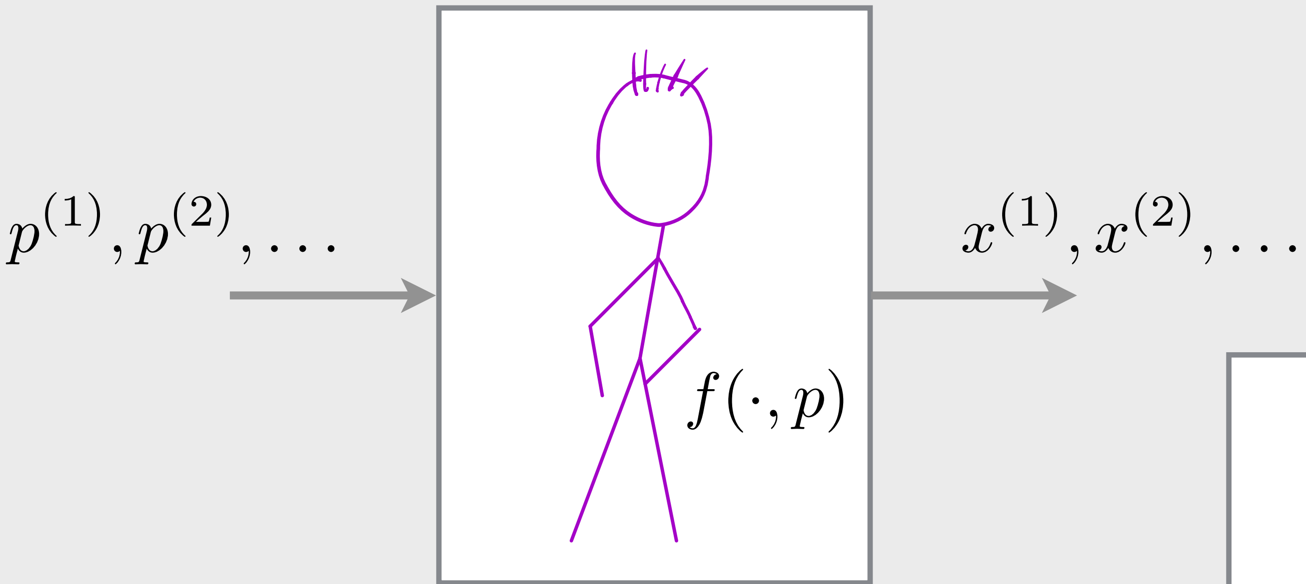
- What if the points $x^{(k)}$ are only **approximately** optimal for every parameter $p^{(k)}$?
- for example, what if the "optimizer" is a human?



we wish to **learn** the expert's objective from demonstrations, so we can **mimic** them later

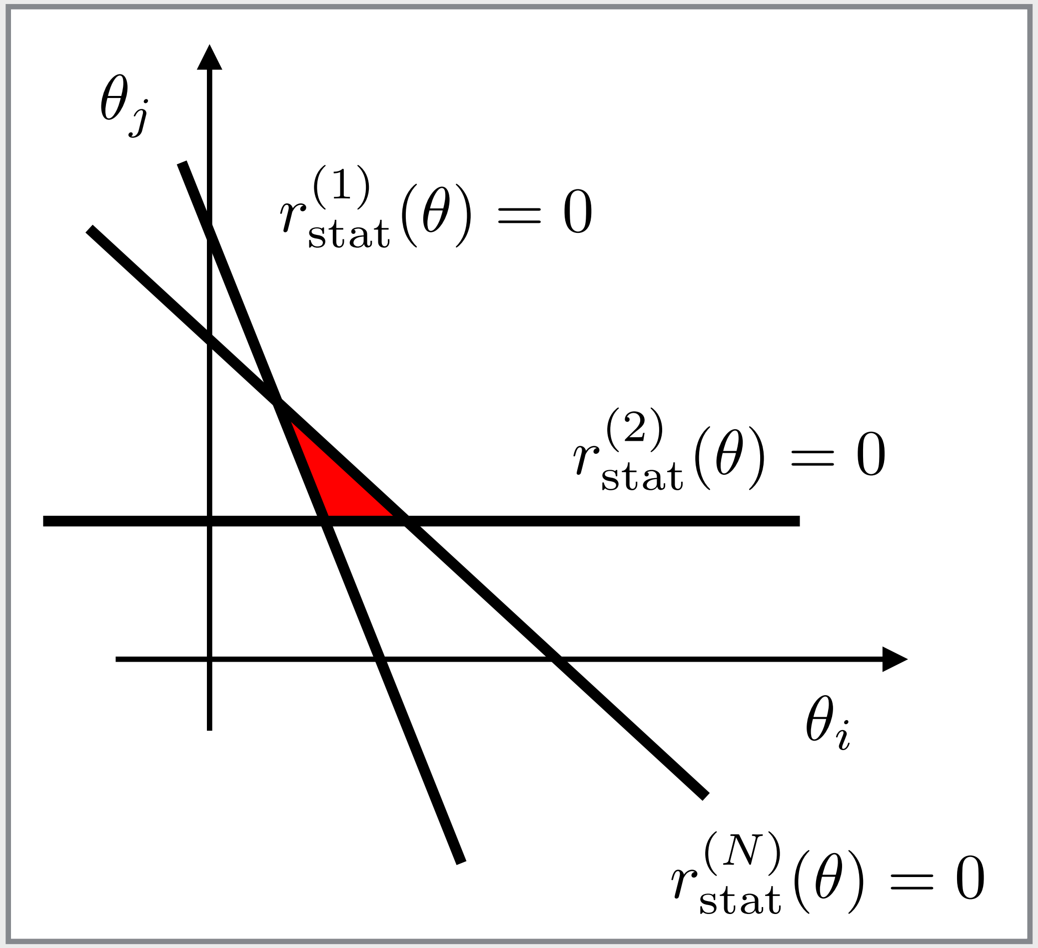
Typical picture

- What if the points $x^{(k)}$ are only **approximately** optimal for every parameter $p^{(k)}$?
- for example, what if the "optimizer" is a human?



we wish to **learn** the expert's objective from demonstrations, so we can **mimic** them later

Human expert can be **suboptimal**, or **inconsistent**



Imputing a minimally inconsistent objective

Convex optimization problem:

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{k=1}^{N} \psi(r_{\mathrm{stat}}^{(k)}(\theta)) \\
\text{subject to} & \theta \in \Theta
\end{array}
$$

Imputing a minimally inconsistent objective

Convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \psi(r_{\text{stat}}^{(k)}(\theta)) \\ \text{subject to} & \theta \in \Theta \end{array}$$

Recall:

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

$$r_{\text{stat}}^{(k)}(\theta) = \sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)})$$

Imputing a minimally inconsistent objective

Convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \psi(r_{\text{stat}}^{(k)}(\theta)) \\ \text{subject to} & \theta \in \Theta \end{array}$$

Recall:

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

$$r_{\text{stat}}^{(k)}(\theta) = \sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)})$$

- minimizes the sum of penalties on the residuals
- l_2 penalty:

$$\psi(r_{\text{stat}}^{(k)}(\theta)) = \|r_{\text{stat}}^{(k)}(\theta)\|_2^2$$

Imputing a minimally inconsistent objective

Convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \psi(r_{\text{stat}}^{(k)}(\theta)) \\ \text{subject to} & \theta \in \Theta \end{array}$$

Recall:

$$\mathcal{D} = \left\{ (x^{(k)}, p^{(k)}) \mid k = 1, \ldots, N \right\}$$

$$r_{\text{stat}}^{(k)}(\theta) = \sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)})$$

- minimizes the sum of penalties on the residuals
- l_2 penalty:

$$\psi(r_{\text{stat}}^{(k)}(\theta)) = \|r_{\text{stat}}^{(k)}(\theta)\|_2^2$$

- the set $\Theta \subseteq \mathbb{R}^M$ incorporates any **prior** knowledge on the basis coefficients
- if the minimum is zero, then
  - **either** the optimal coefficients are consistent with all data points
  - **or** the trivial solution was found (more later)

Discrete formulation

	continuous space	**discrete space**
objective function	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$
domain	$\mathcal{X} \times \mathcal{P} \subseteq \mathbb{R}^n \times \mathbb{R}^M$	$\mathcal{X}$ and $\mathcal{P}$ finite
optimality condition	$\nabla_x f(x^\star, p) = 0$	$f(x^\star, p) \le f(x, p), \text{ for all } x \in \mathcal{X}$
consistency with optimality of *k*th data point	$\sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$	$\sum_{i=1}^{M} \theta_i \left(\phi_i(x, p^{(k)}) - \phi_i(x^{(k)}, p^{(k)})\right) \ge 0, \text{ for all } x \in \mathcal{X}$

Discrete formulation

	continuous space	**discrete space**
objective function	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$
domain	$\mathcal{X} \times \mathcal{P} \subseteq \mathbb{R}^n \times \mathbb{R}^M$	$\mathcal{X}$ and $\mathcal{P}$ finite
optimality condition	$\nabla_x f(x^\star, p) = 0$	$f(x^\star, p) \leq f(x, p), \text{ for all } x \in \mathcal{X}$
consistency with optimality of *k*th data point	$\sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$	$\sum_{i=1}^{M} \theta_i \left(\phi_i(x, p^{(k)}) - \phi_i(x^{(k)}, p^{(k)}) \right) \geq 0,$ for all $x \in \mathcal{X}$

stationarity residual $r_{\text{stat}}^{(k)}(\theta)$

Discrete formulation

	continuous space	**discrete space**
objective function	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$
domain	$\mathcal{X} \times \mathcal{P} \subseteq \mathbb{R}^n \times \mathbb{R}^M$	$\mathcal{X}$ and $\mathcal{P}$ finite
optimality condition	$\nabla_x f(x^\star, p) = 0$	$f(x^\star, p) \leq f(x, p), \text{ for all } x \in \mathcal{X}$
consistency with optimality of *k*th data point	$\sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$	$\sum_{i=1}^{M} \theta_i \left(\phi_i(x, p^{(k)}) - \phi_i(x^{(k)}, p^{(k)}) \right) \geq 0, \text{ for all } x \in \mathcal{X}$
	stationarity residual $r_{\text{stat}}^{(k)}(\theta)$	*consistency* residual $r_{\text{cons}}^{(k)}(x, \theta)$

Discrete formulation

	continuous space	**discrete space**
objective function	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$	$f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$
domain	$\mathcal{X} \times \mathcal{P} \subseteq \mathbb{R}^n \times \mathbb{R}^M$	$\mathcal{X}$ and $\mathcal{P}$ finite
optimality condition	$\nabla_x f(x^\star, p) = 0$	$f(x^\star, p) \leq f(x, p), \ \text{for all } x \in \mathcal{X}$
consistency with optimality of *k*th data point	$\sum_{i=1}^{M} \theta_i \nabla_x \phi(x^{(k)}, p^{(k)}) = 0$	$\sum_{i=1}^{M} \theta_i \left(\phi_i(x, p^{(k)}) - \phi_i(x^{(k)}, p^{(k)})\right) \geq 0, \quad \text{for all } x \in \mathcal{X}$
	***stationarity* residual** $r_{\text{stat}}^{(k)}(\theta)$	***consistency* residual** $r_{\text{cons}}^{(k)}(x, \theta)$
imputing an objective	$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \psi(r_{\text{stat}}^{(k)}(\theta)) \\ \text{subject to} & \theta \in \Theta \end{array}$	$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{ \left(-r_{\text{cons}}^{(k)}(x, \theta)\right)_+ \right\} \\ \text{subject to} & \theta \in \Theta \end{array}$
	convex problem	convex problem

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem parameters
ℓ, A, B, x_0

$$\begin{array}{ll} \text{minimize} & \displaystyle\int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

optimal trajectories
$x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & \displaystyle\int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

→ optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & \displaystyle\int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

← optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & f(x,p) \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

→ optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & \int_0^\infty \ell(x,u)\, dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

← optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & f(x,p) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

→ optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

problem parameters ℓ, A, B, x_0 →

$$\begin{array}{ll} \text{minimize} & \displaystyle\int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

← optimal trajectories $x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem
$p \in \mathcal{P}$

$$\begin{array}{ll} \text{minimize} & f(x,p) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

optimal trajectories
$x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

problem
parameters
ℓ, A, B, x_0

$$\begin{array}{ll} \text{minimize} & \int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

optimal trajectories
$x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem

$p \in \mathcal{P}$

$$\begin{array}{ll} \text{minimize} & f(x,p) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

ectories

$x^\star \in \mathcal{X}$

Inverse optimal control

problem
parameters
ℓ, A, B, x_0

$$\begin{array}{ll} \text{minimize} & \int_0^\infty \ell(x,u)\,dt \\ \text{subject to} & \dot{x} = Ax + Bu \\ & x(0) = x_0 \end{array}$$

optimal trajectories
$x^\star(\cdot), u^\star(\cdot)$

Inverse optimal control

- in control: studied by Kalman (1964), Boyd (1994) and others
- in ML: sometimes called **inverse reinforcement learning**
  Ng & Russell (2000), Abbeel & Ng (2004), Abbeel, Coates et al (2010)

Forward optimal control

problem

$p \in \mathcal{P}$ →

minimize	$f(x,p)$
subject to	$x \in \mathcal{X}$

→ ectories

$x^\star \in \mathcal{X}$

Inverse optimal control

problem
parameters
ℓ, A, B, x_0 →

minimize	$f = \sum_i \theta_i \phi_i$
subject to	$x = Ax + Bu$
	$x(0) = x_0$

← optimal trajectories
$x^\star(\cdot), u^\star(\cdot)$

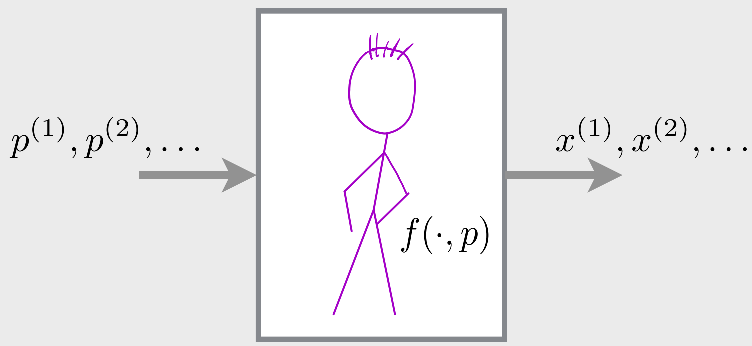
Reward hypothesis

…all of what we mean by goals and purposes can be well thought of as the maximization [resp. minimization] of … a received scalar signal (called *reward* [*stage cost*]) (Sutton & Barto 1998)

Learning by expert demonstration

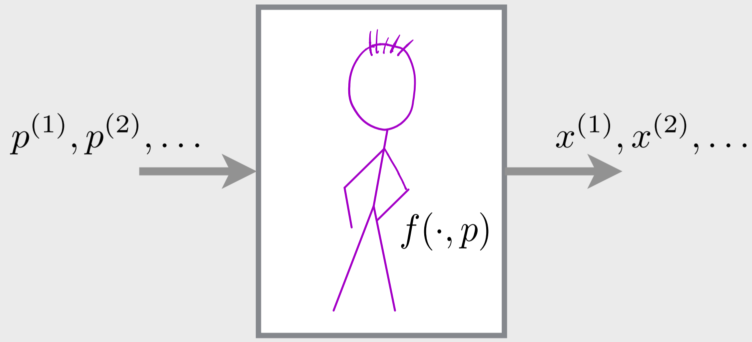
Learning by expert demonstration

1. Expert gives "optimal" demonstrations

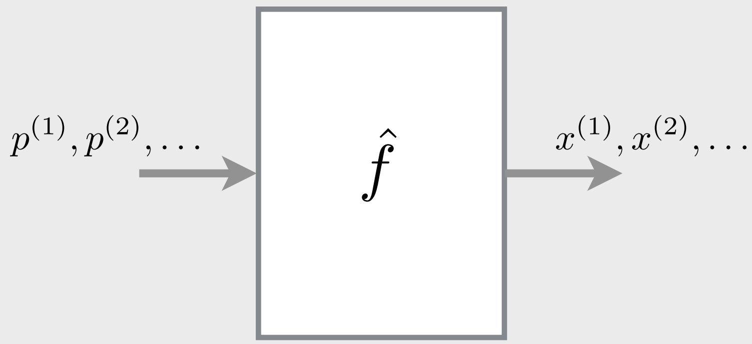


Learning by expert demonstration

1. Expert gives “optimal” demonstrations

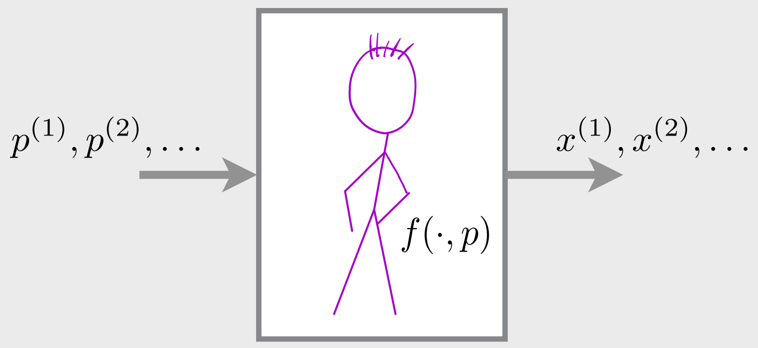


2. Demonstrations used to “learn” the expert

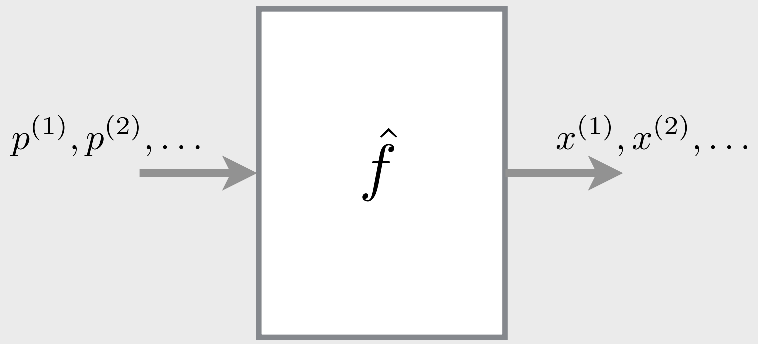


Learning by expert demonstration

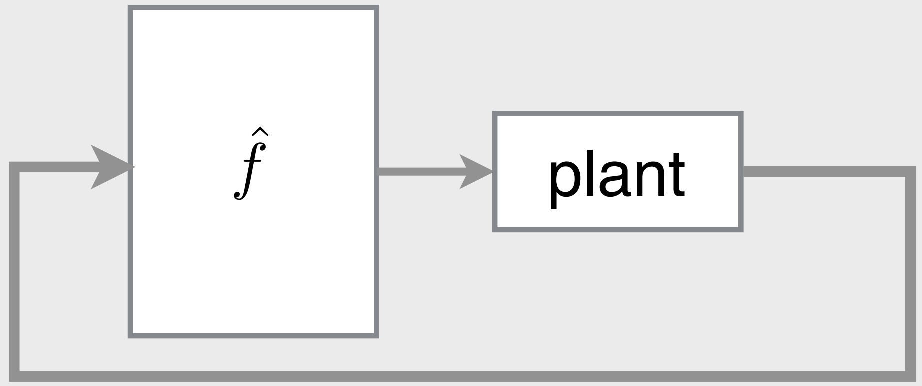
1. Expert gives “optimal” demonstrations



2. Demonstrations used to “learn” the expert

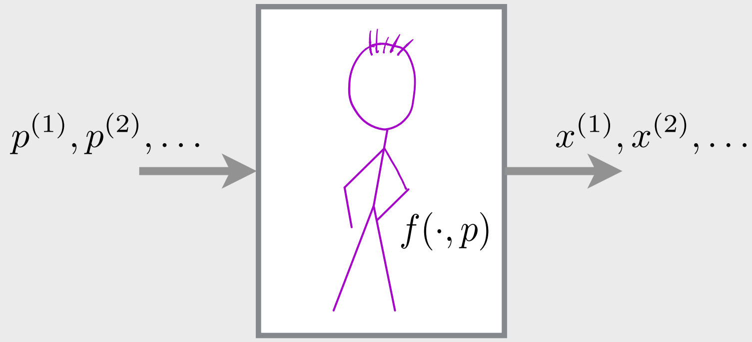


3. Learned objective (e.g. loss function) used to mimic the expert in an autonomous system

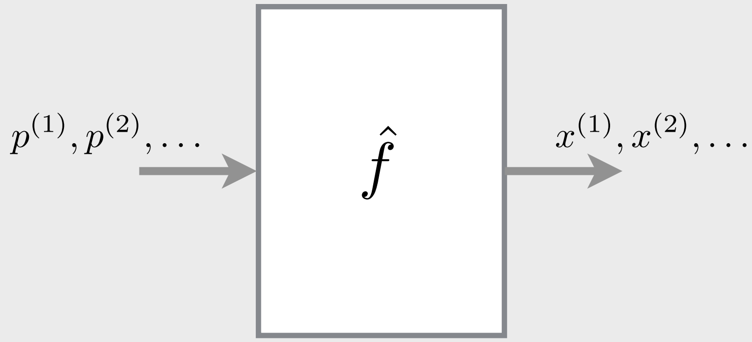


Learning by expert demonstration

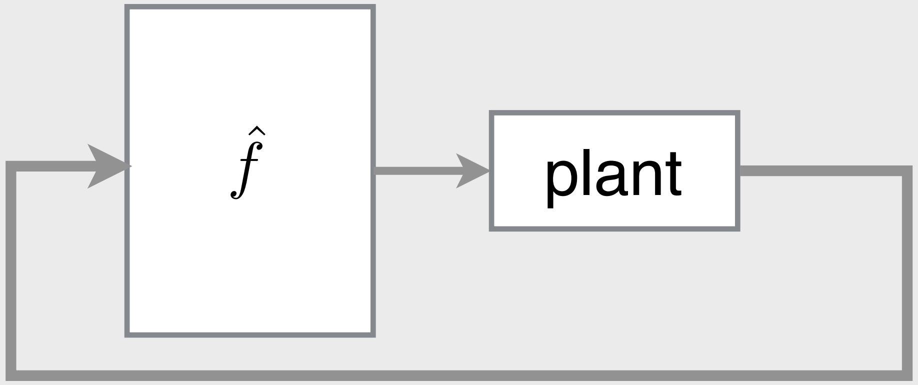
1. Expert gives "optimal" demonstrations



2. Demonstrations used to "learn" the expert



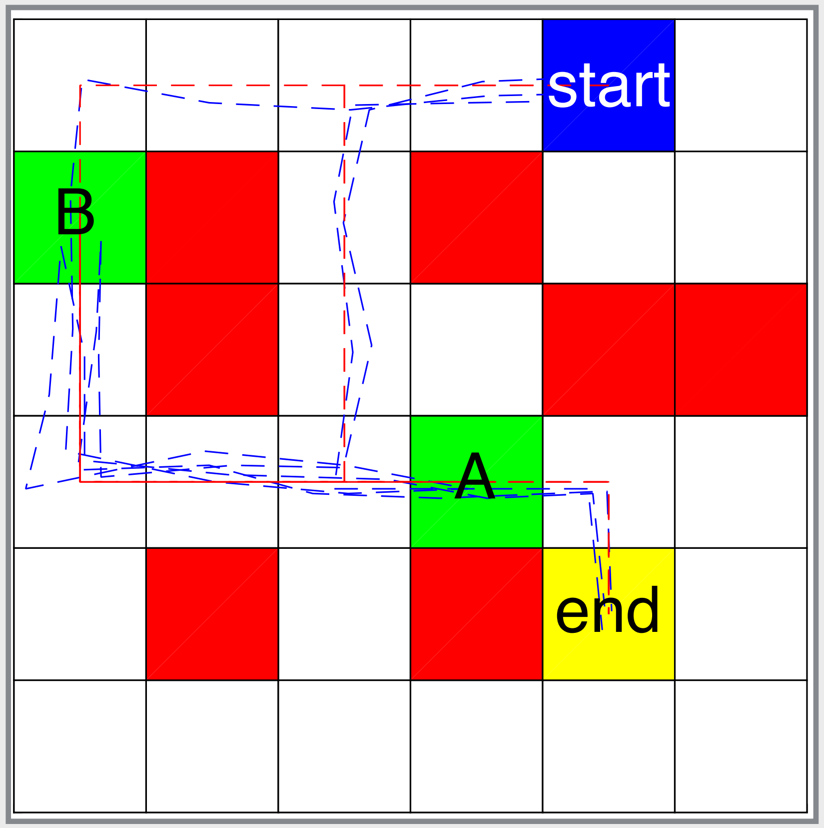
3. Learned objective (e.g. loss function) used to mimic the expert in an autonomous system



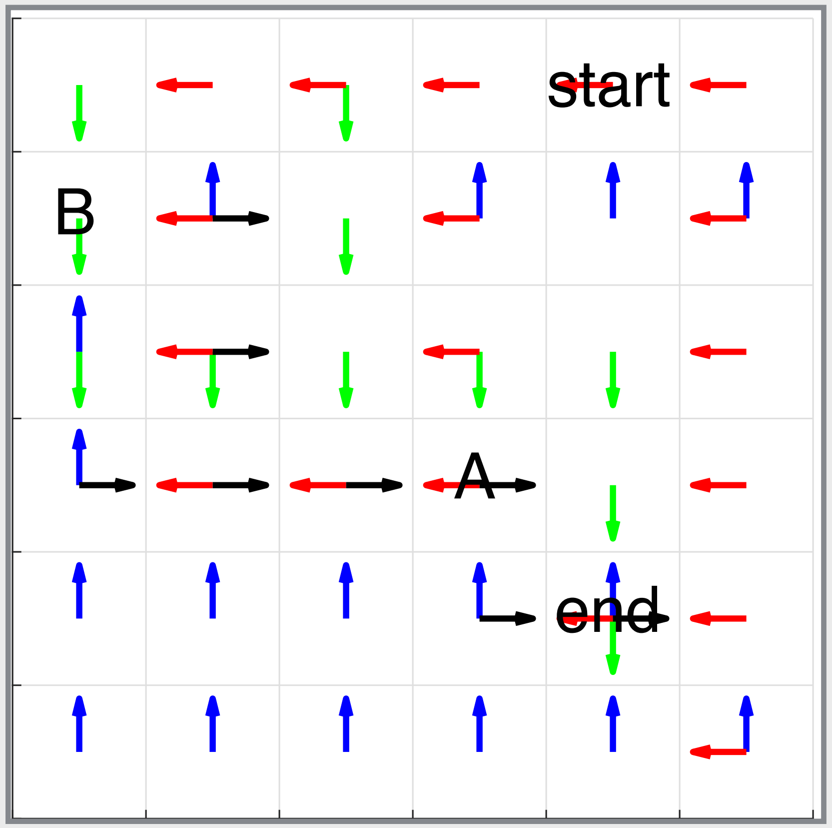
heli.stanford.edu

Learning fails when the expert is inconsistent

Expert demonstrations:



Learned policy:



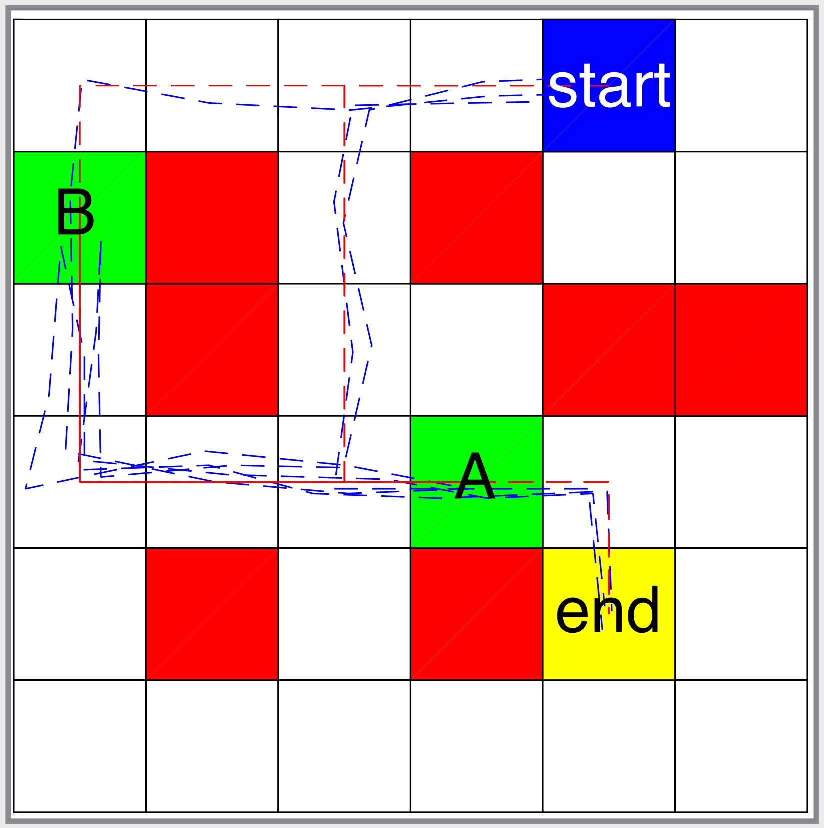
Task: get from start to end in the fewest steps, while visiting A and B in any order

- inverse optimal control applied to a grid world
- dynamics are modeled as a transition system
- learned an approximation to the optimal value function $V^\star(s)$

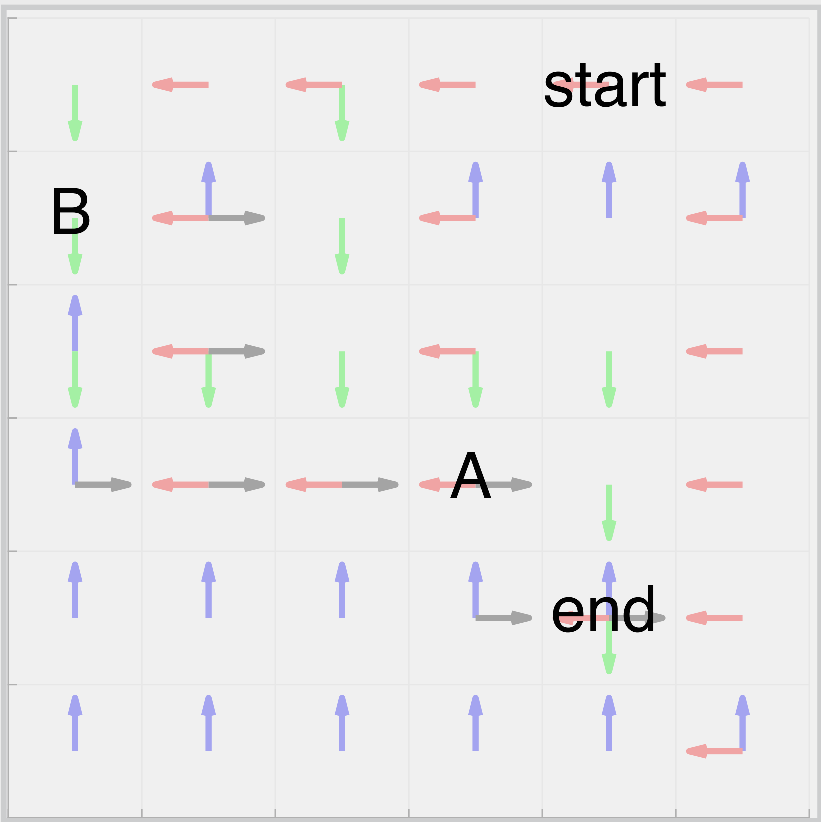
$$f(x,p) = \ell(s,s') + \hat{V}(s')$$

Learning fails when the expert is inconsistent

Expert demonstrations:



Learned policy:



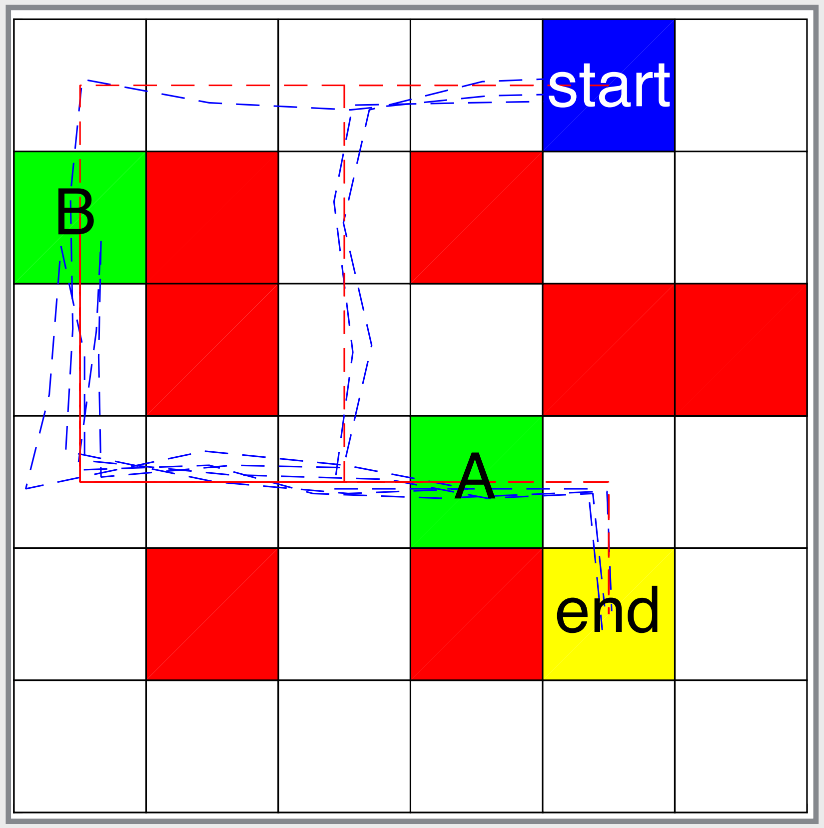
Task: get from start to end in the fewest steps, while visiting A and B in any order

- inverse optimal control applied to a grid world
- dynamics are modeled as a transition system
- learned an approximation to the optimal value function $V^\star(s)$

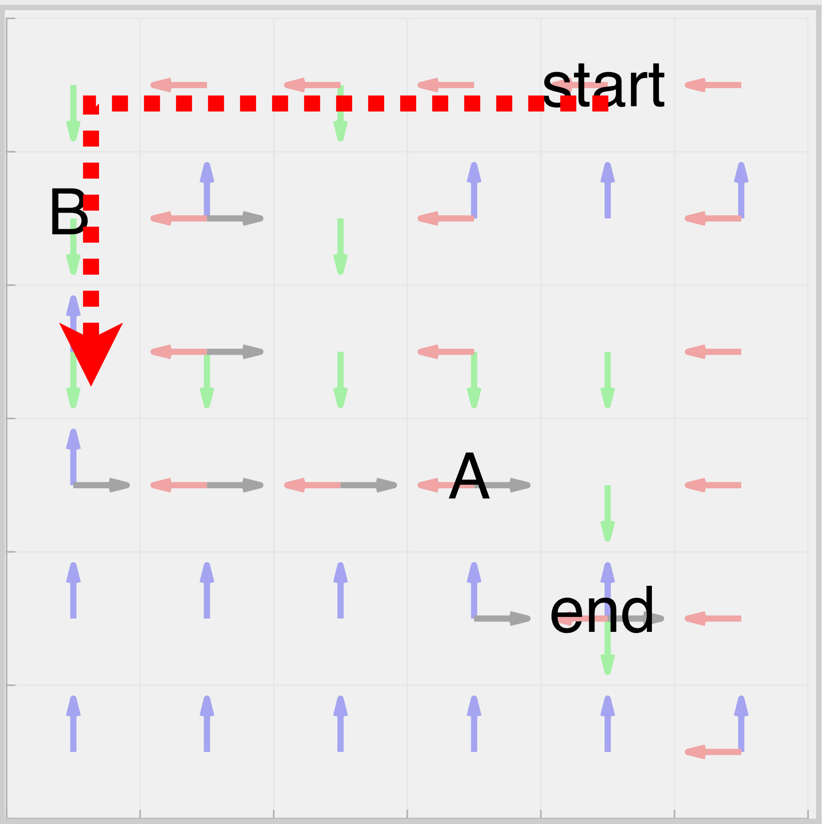
$$f(x,p) = \ell(s,s') + \hat{V}(s')$$

Learning fails when the expert is inconsistent

Expert demonstrations:



Learned policy:



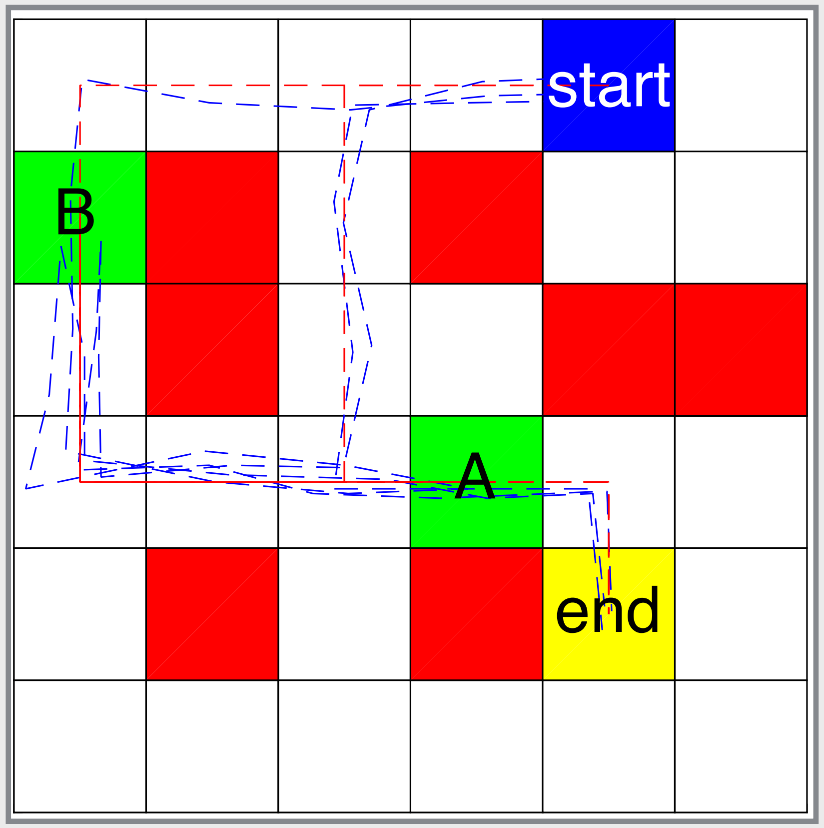
Task: get from start to end in the fewest steps, while visiting A and B in any order

- inverse optimal control applied to a grid world
- dynamics are modeled as a transition system
- learned an approximation to the optimal value function $V^{\star}(s)$

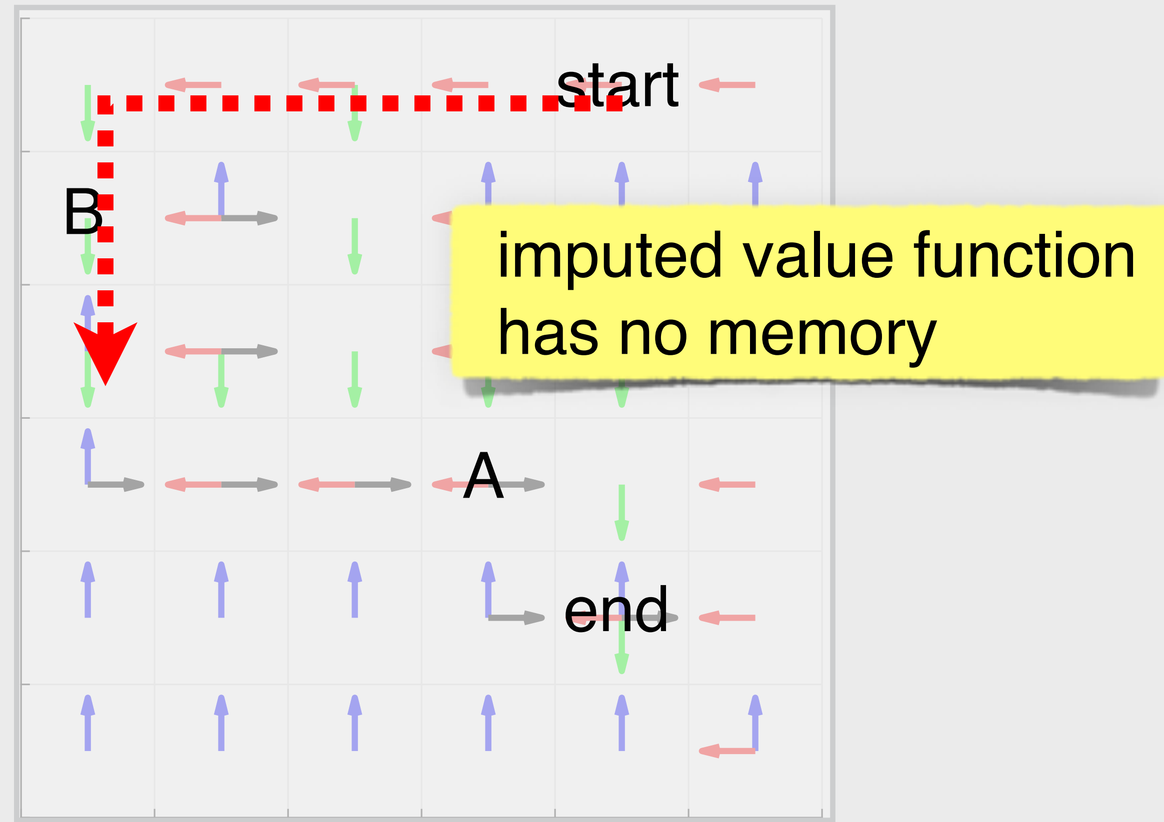
$$f(x, p) = \ell(s, s') + \hat{V}(s')$$

Learning fails when the expert is inconsistent

Expert demonstrations:



Learned policy:



Task: get from start to end in the fewest steps, while visiting A and B in any order

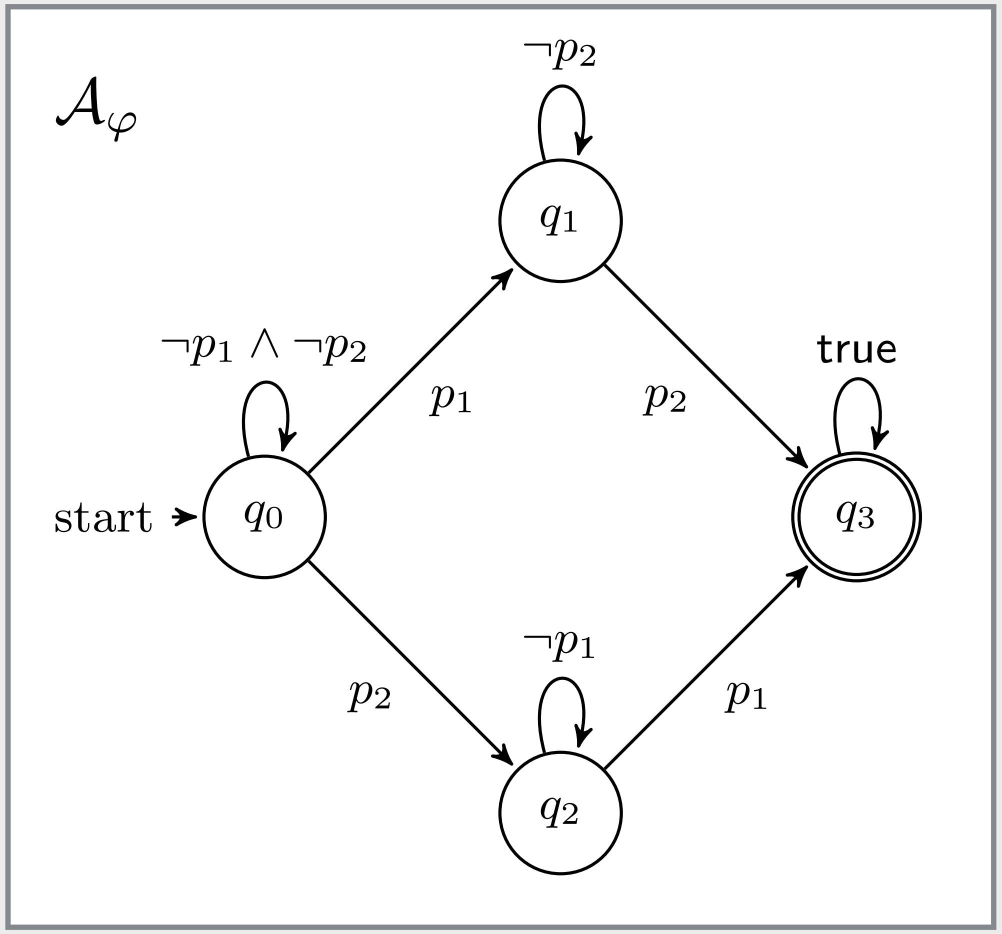
- inverse optimal control applied to a grid world
- dynamics are modeled as a transition system
- learned an approximation to the optimal value function $V^\star(s)$

$$f(x,p) = \ell(s,s') + \hat{V}(s')$$

Side information

Task: get from start to end in the fewest steps, while visiting A and B in any order

Side information automaton



time	0	1	2	3	…	t	t+1
state	s_1	s_2	s_3	s_4	…	s_t	s_{t+1}
p_1	F	F	T	F		T	F
p_2	F	F	F	F		T	F

Side information: a specification automaton that every "optimal" trajectory must satisfy.

At each time step the atomic propositions p_1 and p_2 are evaluated

- p_1 = true iff. the state is A
- p_2 = true iff. the state is B

In the inverse problem, the side information becomes a **hidden state** with (known) evolution

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \underset{\{\alpha \mid s \xrightarrow{\alpha} s'\}}{\operatorname{argmin}} \left\{ \ell(s,\alpha,s') + V_{t+1}^\star(s') \right\}$$

Mode-varying policy:

$$\mu^\star(s,q) \in \underset{\{\alpha \mid s \xrightarrow{\alpha} s', q' = \delta(q, L(s))\}}{\operatorname{argmin}} \{ \ell(s,\alpha,s') + V^\star(s',q') \}$$

Expert data:

$$\mathcal{D} = \left\{ \left((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)}) \right) \mid k = 1, \ldots, N \right\}$$

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \underset{\{\alpha | s \xrightarrow{\alpha} s'\}}{\operatorname{argmin}} \left\{\ell(s,\alpha,s') + V_{t+1}^\star(s')\right\}$$

optimizing process

Mode-varying policy:

$$\mu^\star(s,q) \in \underset{\{\alpha | s \xrightarrow{\alpha} s', q'=\delta(q,L(s))\}}{\operatorname{argmin}} \{\ell(s,\alpha,s') + V^\star(s',q')\}$$

Expert data:

$$\mathcal{D} = \left\{\left((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)})\right) \mid k = 1, \ldots, N\right\}$$

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \underset{\{\alpha | s \xrightarrow{\alpha} s'\}}{\operatorname{argmin}} \{\ell(s, \alpha, s') + V_{t+1}^\star(s')\}$$

optimizing process

learned by IOC

Mode-varying policy:

$$\mu^\star(s, q) \in \underset{\{\alpha | s \xrightarrow{\alpha} s', q' = \delta(q, L(s))\}}{\operatorname{argmin}} \{\ell(s, \alpha, s') + V^\star(s', q')\}$$

Expert data:

$$\mathcal{D} = \left\{ \left((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)}) \right) \mid k = 1, \ldots, N \right\}$$

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \underset{\{\alpha | s \xrightarrow{\alpha} s'\}}{\operatorname{argmin}} \left\{ \ell(s, \alpha, s') + V_{t+1}^\star(s') \right\}$$

learned by IOC

optimizing process

Mode-varying policy:

$$\mu^\star(s, q) \in \underset{\{\alpha | s \xrightarrow{\alpha} s', q' = \delta(q, L(s))\}}{\operatorname{argmin}} \{ \ell(s, \alpha, s') + V^\star(s', q') \}$$

Expert data:

$$\mathcal{D} = \left\{ \left((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)}) \right) \mid k = 1, \ldots, N \right\}$$

$x^{(k)}$

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \operatorname*{argmin}_{\{\alpha \,|\, s \xrightarrow{\alpha} s'\}} \left\{ \ell(s,\alpha,s') + V_{t+1}^\star(s') \right\}$$

learned by IOC

optimizing process

Mode-varying policy:

$$\mu^\star(s,q) \in \operatorname*{argmin}_{\{\alpha \,|\, s \xrightarrow{\alpha} s',\, q' = \delta(q, L(s))\}} \left\{ \ell(s,\alpha,s') + V^\star(s',q') \right\}$$

Expert data:

$$\mathcal{D} = \left\{ \Big((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)}) \Big) \mid k = 1, \ldots, N \right\}$$

$x^{(k)}$ $\quad$ $p^{(k)}$

Data structures

Memoryless policy:

$$\mu_t^\star(s) \in \operatorname*{argmin}_{\{\alpha \mid s \xrightarrow{\alpha} s'\}} \{\ell(s,\alpha,s') + V_{t+1}^\star(s')\}$$

optimizing process

learned by IOC

Mode-varying policy:

$$\mu^\star(s,q) \in \operatorname*{argmin}_{\{\alpha \mid s \xrightarrow{\alpha} s',\, q'=\delta(q,L(s))\}} \{\ell(s,\alpha,s') + V^\star(s',q')\}$$

Expert data:

$$\mathcal{D} = \Big\{ \Big((\alpha^{(k)}, s'^{(k)}, q'^{(k)}),\ (s^{(k)}, q^{(k)}) \Big) \mid k = 1,\ldots,N \Big\}$$

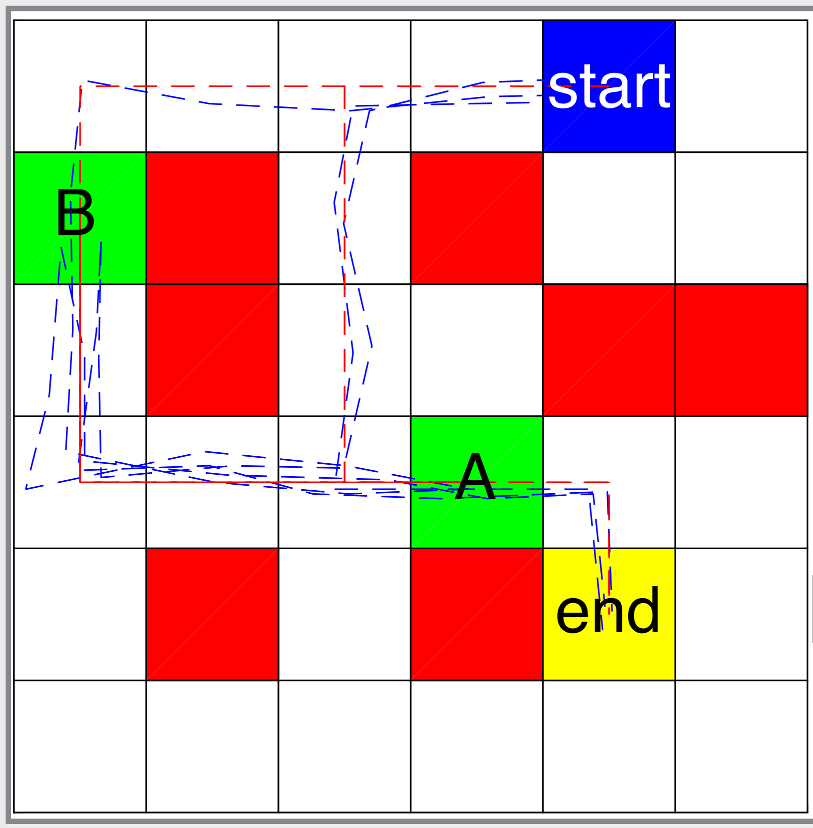
$x^{(k)}$ $p^{(k)}$

compare to memoryless case:

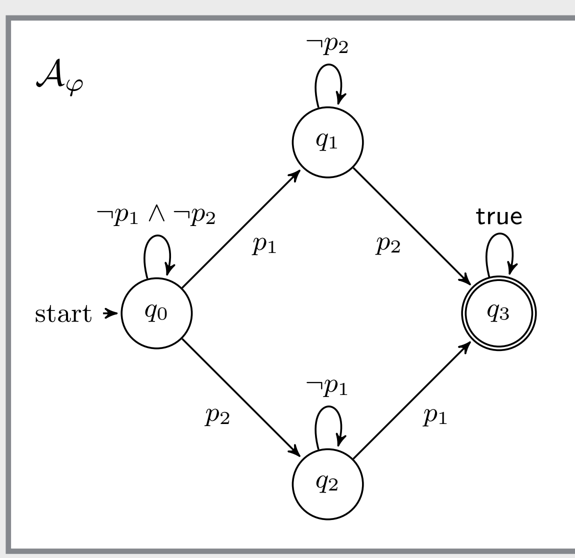
$$\mathcal{D} = \Big\{ \Big((\alpha^{(k)}, s'^{(k)}),\ s^{(k)} \Big) \mid k = 1,\ldots,N \Big\}$$

Side information-aware policy has memory

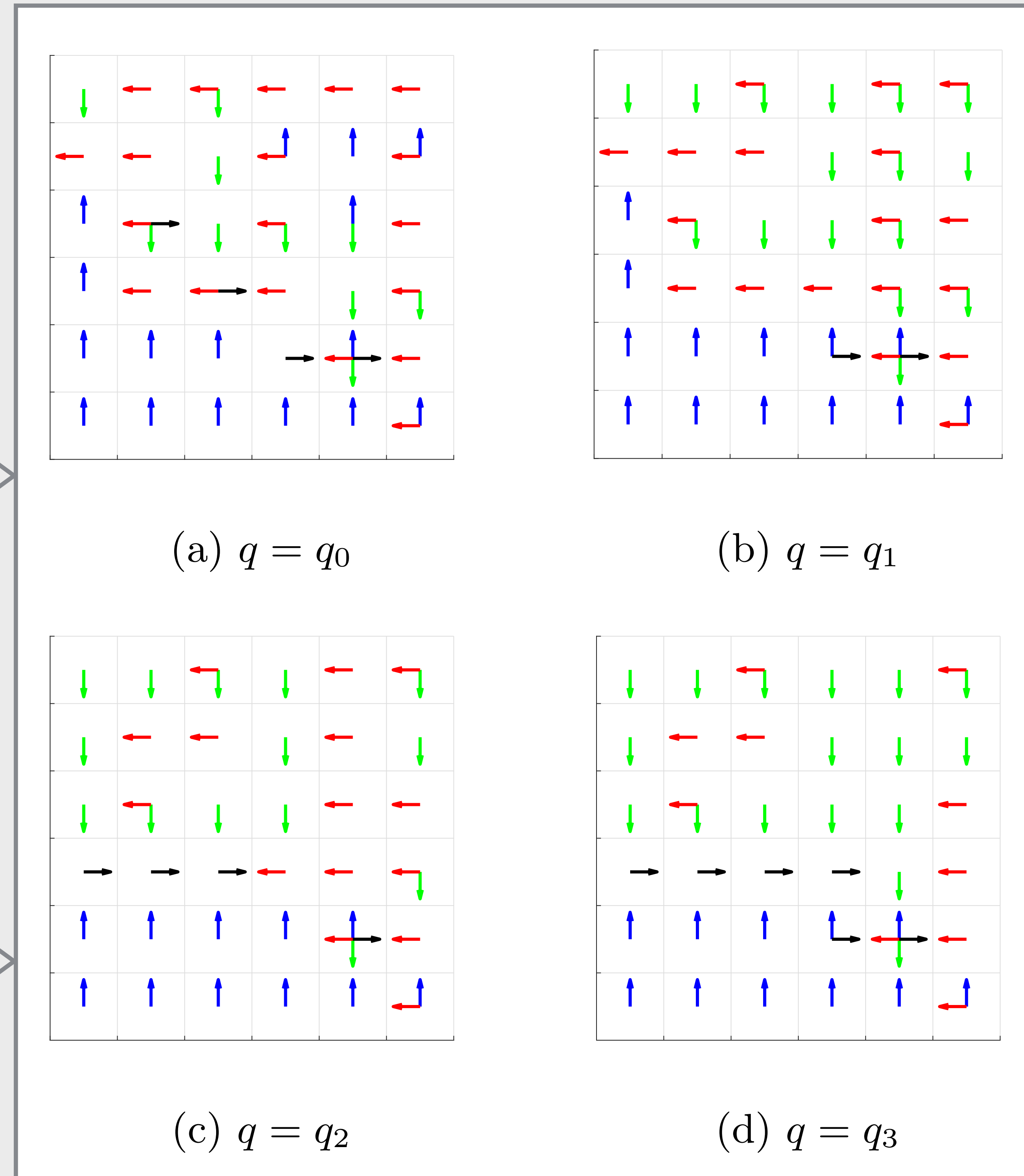
Expert demonstrations:



Side information:

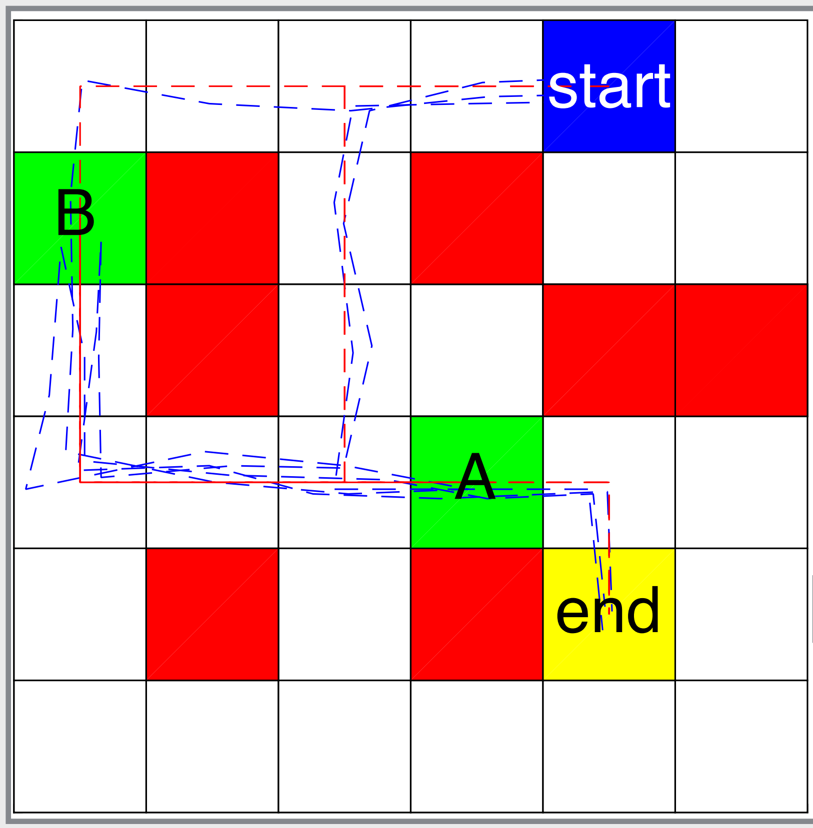


Learned policy:

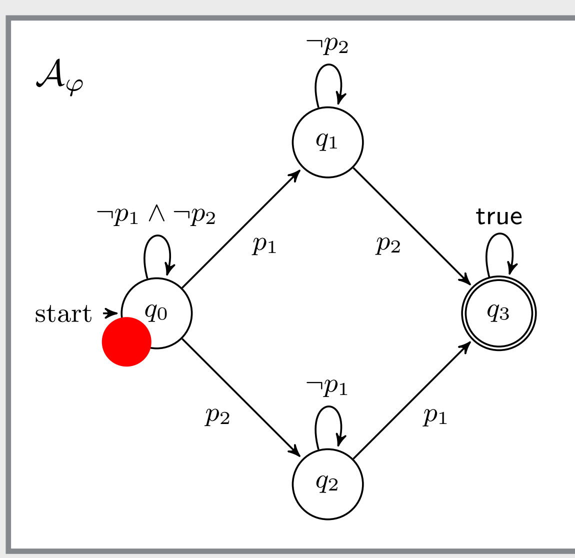


Side information-aware policy has memory

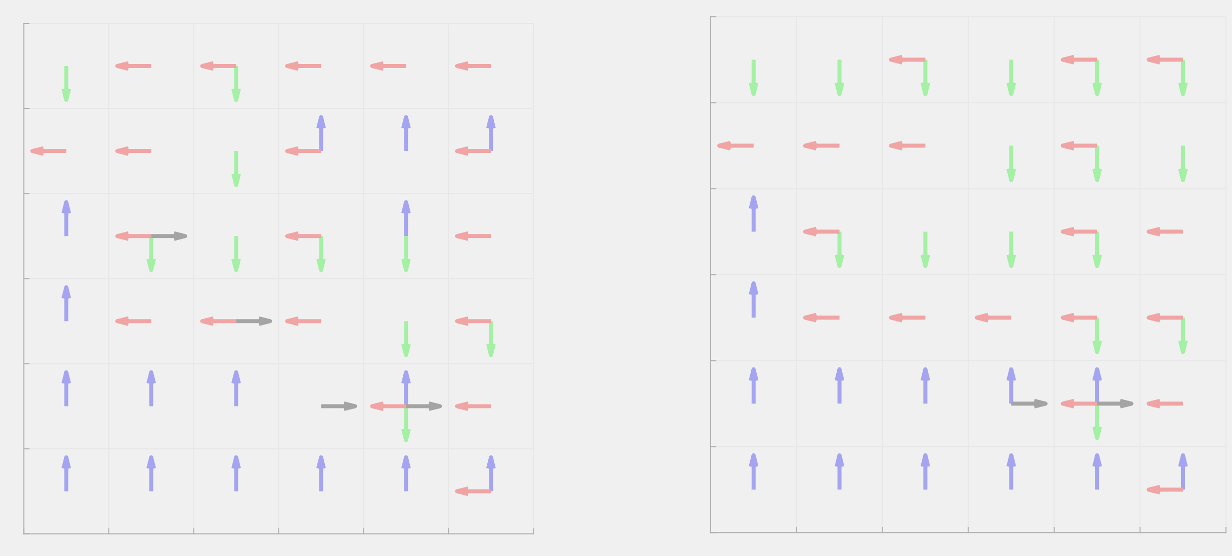
Expert demonstrations:



Side information:



Learned policy:



(a) $q = q_0$

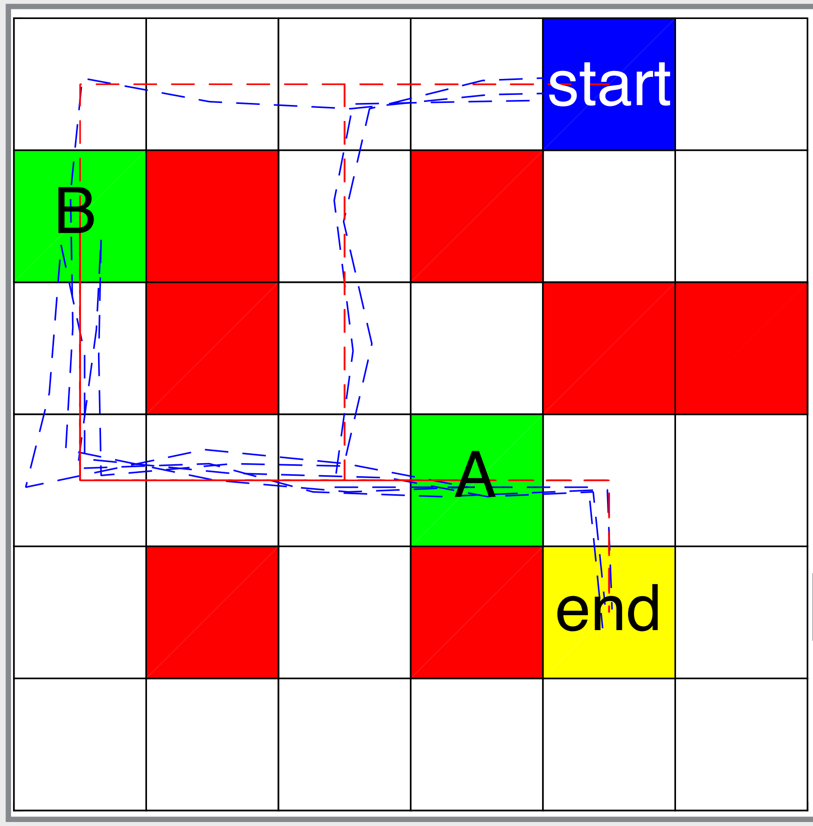
(b) $q = q_1$

(c) $q = q_2$

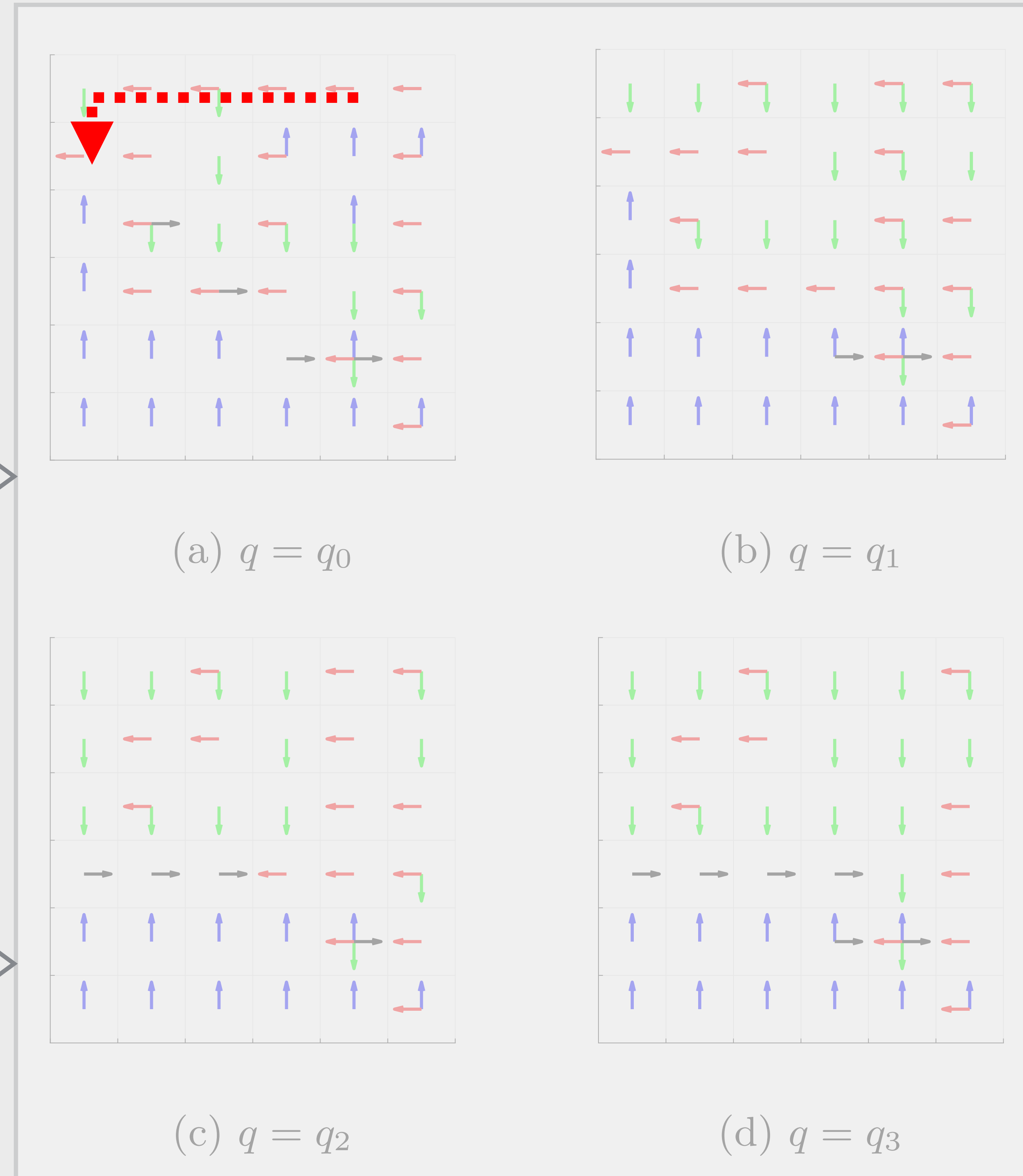
(d) $q = q_3$

Side information-aware policy has memory

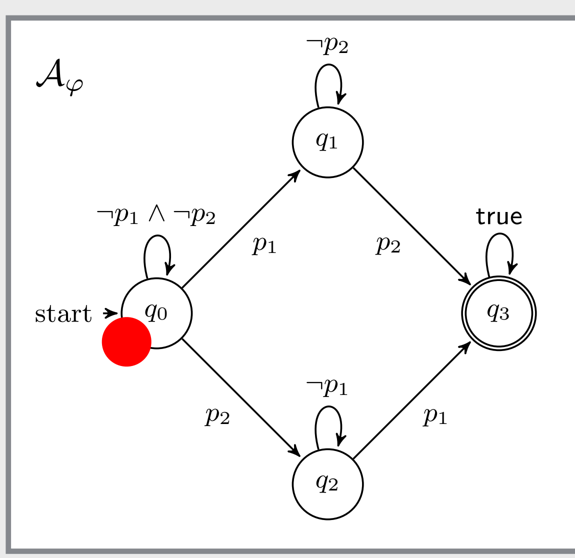
Expert demonstrations:



Learned policy:

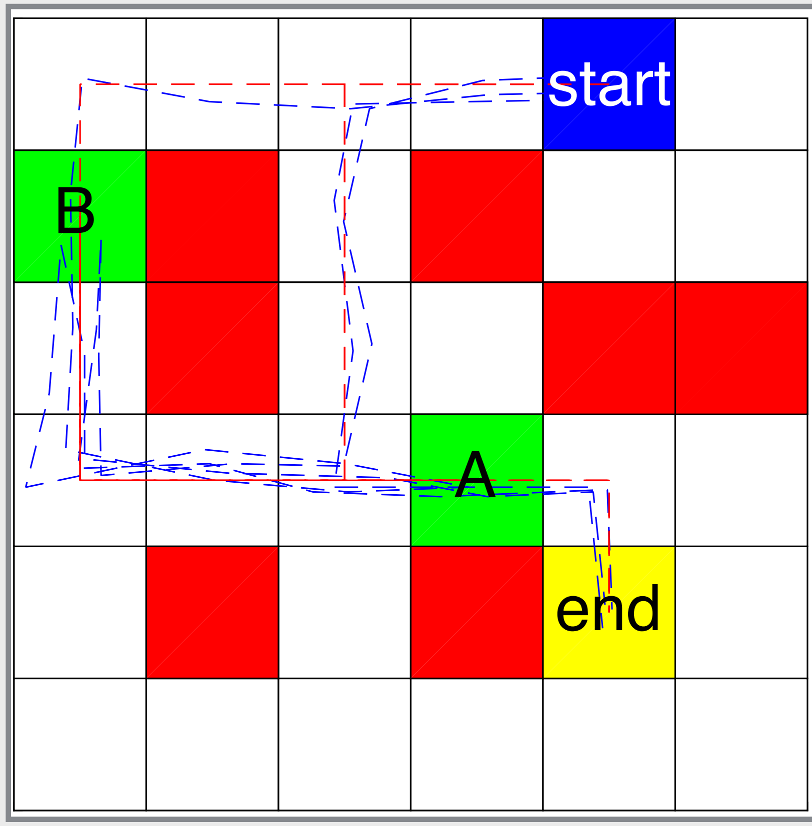


Side information:

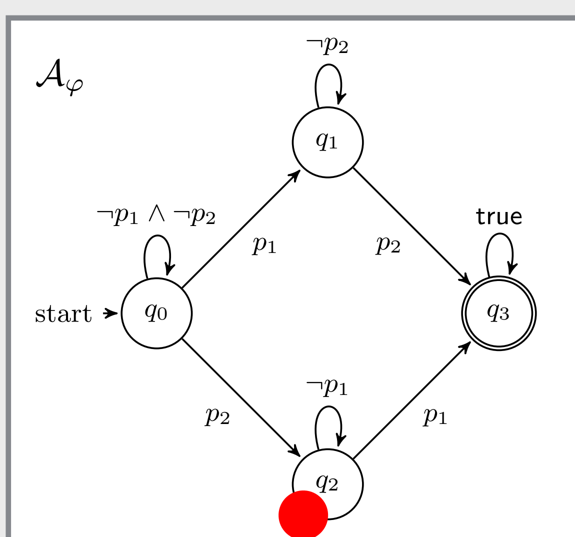


Side information-aware policy has memory

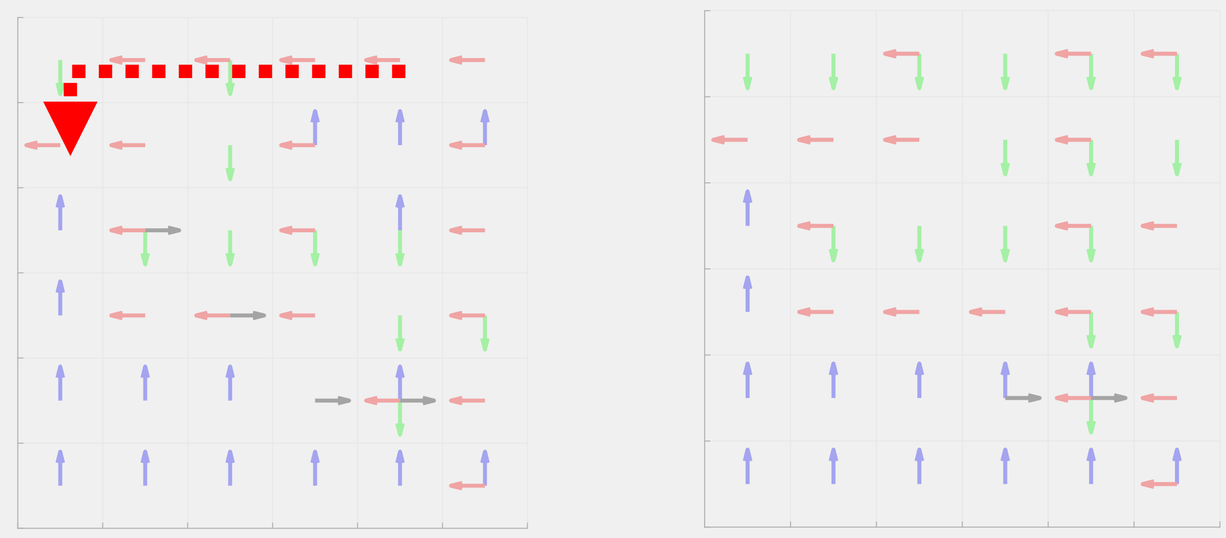
Expert demonstrations:



Side information:



Learned policy:



(a) $q = q_0$

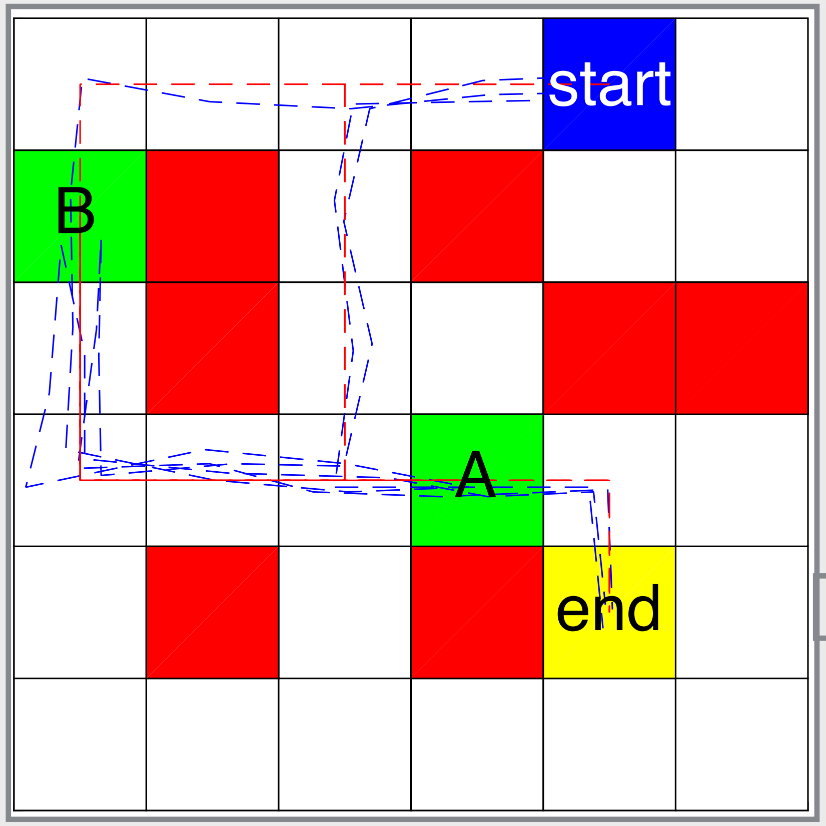
(b) $q = q_1$

(c) $q = q_2$

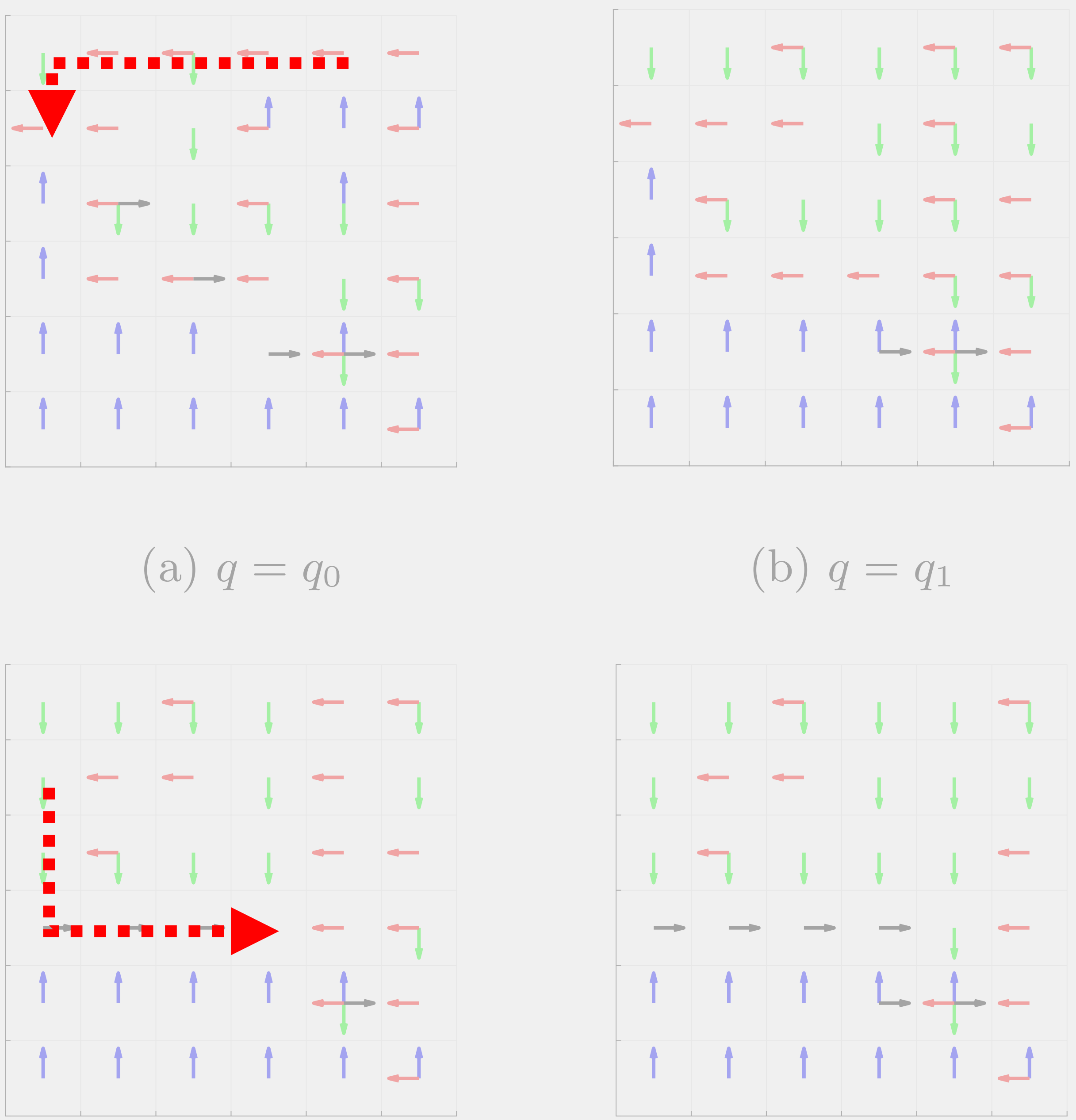
(d) $q = q_3$

Side information-aware policy has memory

Expert demonstrations:



Learned policy:



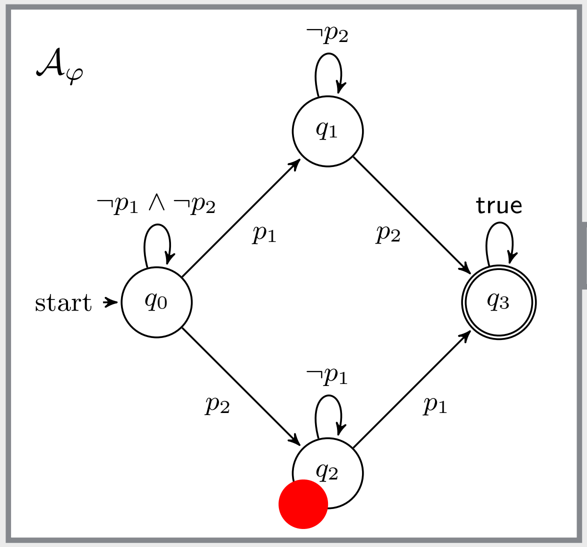
(a) $q = q_0$

(b) $q = q_1$

(c) $q = q_2$

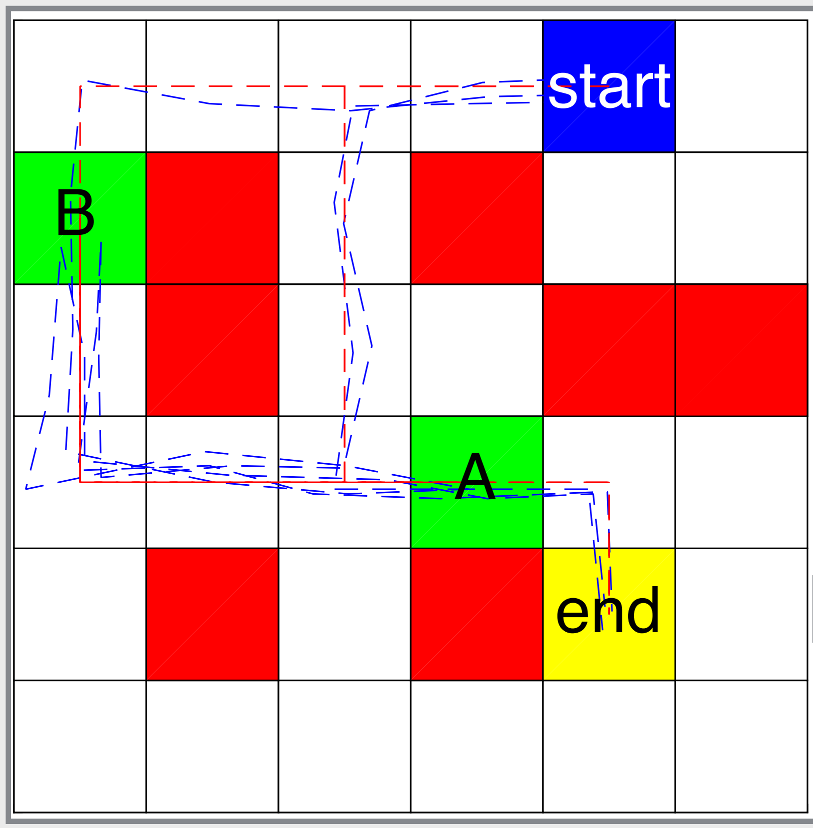
(d) $q = q_3$

Side information:

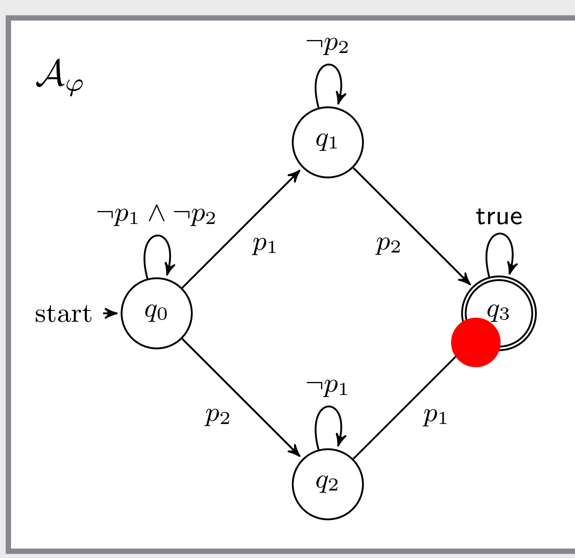


Side information-aware policy has memory

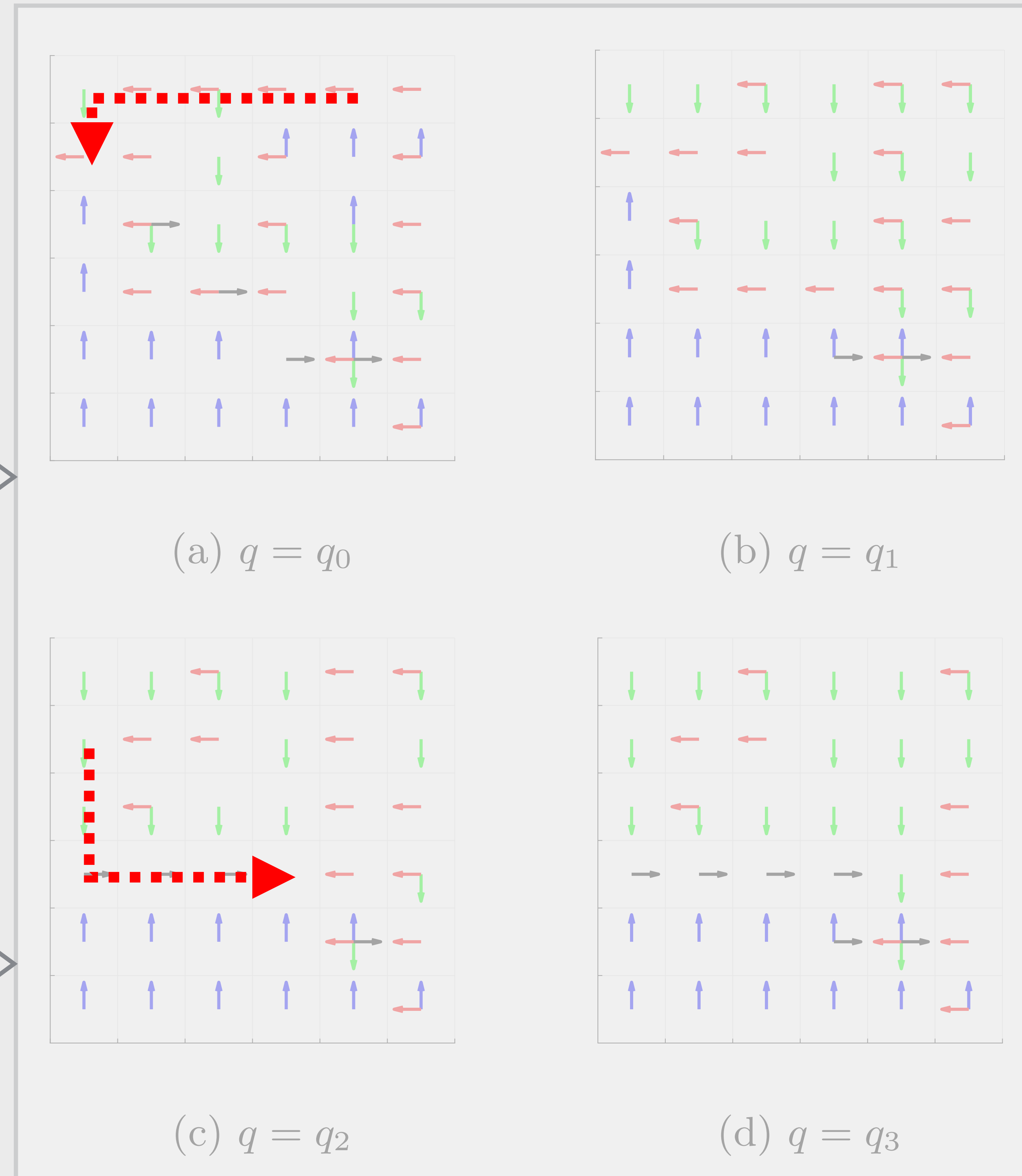
Expert demonstrations:



Side information:

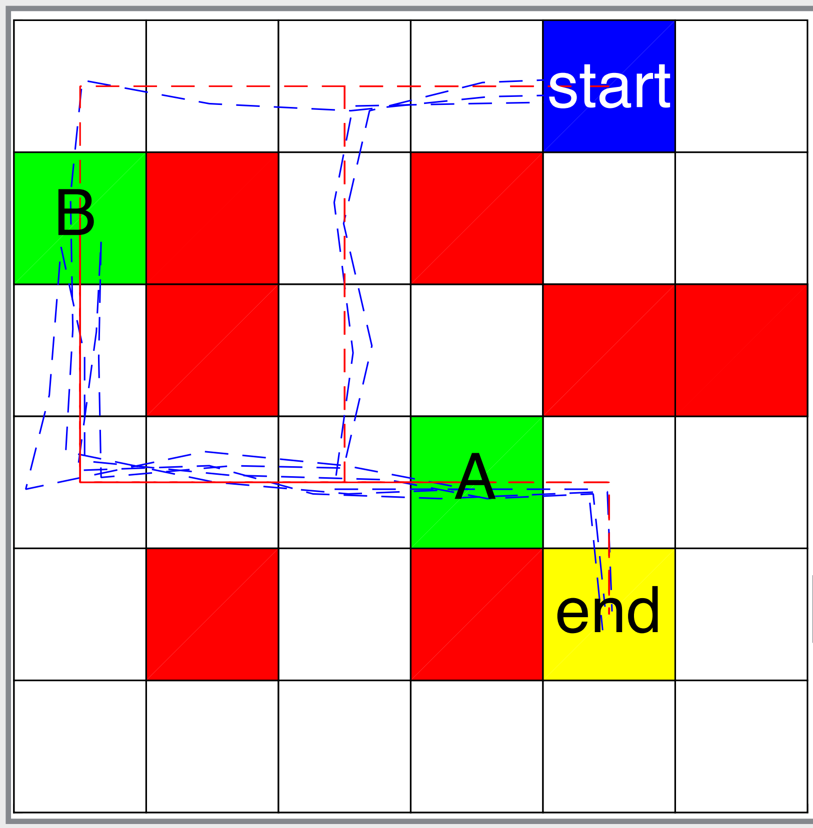


Learned policy:

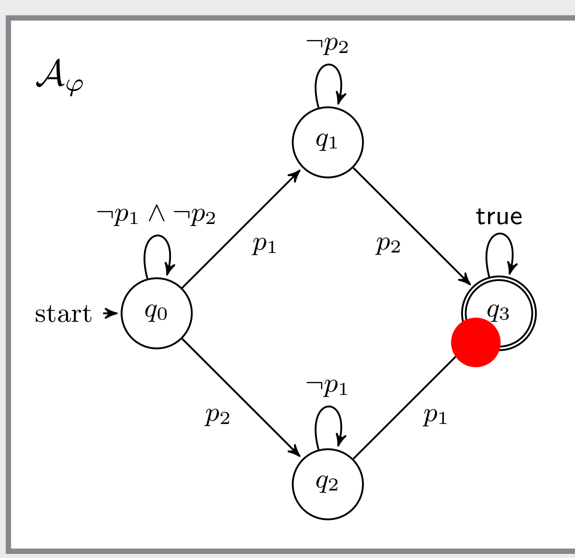


Side information-aware policy has memory

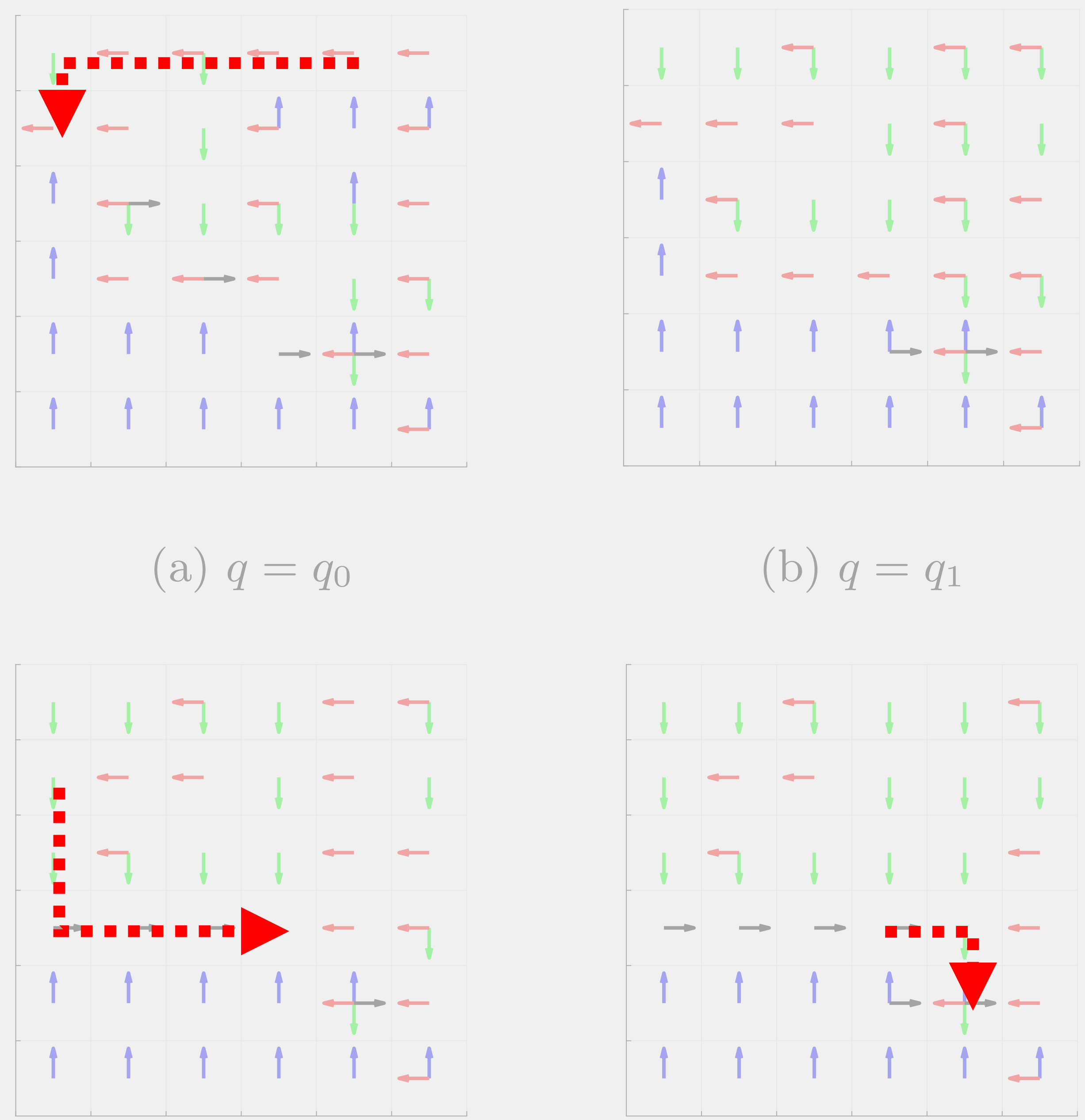
Expert demonstrations:



Side information:



Learned policy:



(a) $q = q_0$ (b) $q = q_1$

(c) $q = q_2$ (d) $q = q_3$

Expert data derivability

inverse optimal control:
(convex problem)

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{ \left(-r_{\text{cons}}^{(k)}(x, \theta) \right)_+ \right\} \\ \text{subject to} & \theta \in \Theta \end{array}$$

→ approximate value function

Expert data derivability

required expert data:

$$\mathcal{D} = \left\{ \Big((\alpha^{(k)}, s'^{(k)}, q'^{(k)}),\ (s^{(k)}, q^{(k)}) \Big) \mid k = 1, \dots, N \right\}$$

inverse optimal control:
(convex problem)

$$\boxed{\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{ \Big(-r^{(k)}_{\text{cons}}(x, \theta) \Big)_+ \right\} \\ \text{subject to} & \theta \in \Theta \end{array}}$$

approximate
value function

Expert data derivability

mode-tracked trajectories:

$$\mathcal{D}' = \left\{ \left(\alpha^{(k)},\ (s^{(k)}, q^{(k)}) \right) \mid k = 1, \ldots, N \right\}$$

$\downarrow$ TS is deterministic

required expert data:

$$\mathcal{D} = \left\{ \left((\alpha^{(k)}, s'^{(k)}, q'^{(k)}),\ (s^{(k)}, q^{(k)}) \right) \mid k = 1, \ldots, N \right\}$$

$\downarrow$

inverse optimal control: (convex problem)

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{ \left(-r_{\text{cons}}^{(k)}(x, \theta) \right)_+ \right\} \\ \text{subject to} & \theta \in \Theta \end{array}$$

$\rightarrow$ approximate value function

Expert data derivability

optimal trajectories: $\mathcal{D}'' = \left\{(\alpha^{(k)},\ s^{(k)}) \mid k = 1, \ldots, N\right\}$

$\downarrow$ + in-step TS and automaton simulation

mode-tracked trajectories: $\mathcal{D}' = \left\{\left(\alpha^{(k)},\ (s^{(k)}, q^{(k)})\right) \mid k = 1, \ldots, N\right\}$

$\downarrow$ TS is deterministic

required expert data: $\mathcal{D} = \left\{\left((\alpha^{(k)}, {s'}^{(k)}, {q'}^{(k)}),\ (s^{(k)}, q^{(k)})\right) \mid k = 1, \ldots, N\right\}$

$\downarrow$

inverse optimal control: (convex problem)

$$\boxed{\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{\left(-r_{\text{cons}}^{(k)}(x, \theta)\right)_+\right\} \\ \text{subject to} & \theta \in \Theta \end{array}}$$

$\rightarrow$ approximate value function

Expert data derivability

only optimal action-states required from expert

optimal trajectories: $\mathcal{D}'' = \left\{(\alpha^{(k)},\ s^{(k)}) \mid k = 1, \ldots, N\right\}$

$\Big\downarrow$ + in-step TS and automaton simulation

mode-tracked trajectories: $\mathcal{D}' = \left\{\left(\alpha^{(k)},\ (s^{(k)}, q^{(k)})\right) \mid k = 1, \ldots, N\right\}$

$\Big\downarrow$ TS is deterministic

required expert data: $\mathcal{D} = \left\{\left((\alpha^{(k)}, s'^{(k)}, q'^{(k)}),\ (s^{(k)}, q^{(k)})\right) \mid k = 1, \ldots, N\right\}$

$\Big\downarrow$

inverse optimal control: (convex problem)

$$\boxed{\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{N} \max_{x \in \mathcal{X}_k} \left\{\left(-r^{(k)}_{\text{cons}}(x, \theta)\right)_+\right\} \\ \text{subject to} & \theta \in \Theta \end{array}}$$

$\longrightarrow$ approximate value function

What's next?

direct extensions

- Extend to broader dynamics classes—hybrid, nonlinear…
- Expand the family of specifications and languages
- Investigate the role of stochastic policies and partially specified side information
- Demonstrate scalability

new opportunities

- Open up a broad set of new problems to ideas from control and optimization

CDC 2016

Automata Theory Meets Approximate Dynamic Programming: Optimal Control with Temporal Logic Constraints

Ivan Papusha[†] Jie Fu[*] Ufuk Topcu[†] Richard M. Murray[†]

HSCC 2017 (under review)

Automata Specifications as Side Information in Inverse Optimal Control

Ivan Papusha
Institute for Computational Engineering and Sciences
University of Texas at Austin
Austin, TX, USA
ipapusha@utexas.edu

Min Wen
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA, USA
wenm@seas.upenn.edu

Ufuk Topcu
Aerospace Engineering and Engineering Mechanics
University of Texas at Austin
Austin, TX, USA
utopcu@utexas.edu