Introduction to Theory of Safe Decision Making

Dr. Akhil Anand

1st AID Scientific Workshop, Trondheim

Forewords & Disclaimer

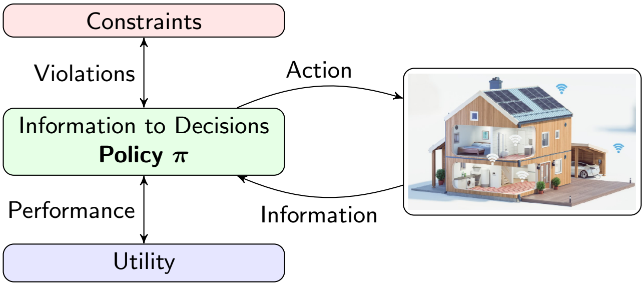
Objectives:

- Put in place some common concepts & language
- Identify some key points in safe decision making
- Connect to AI

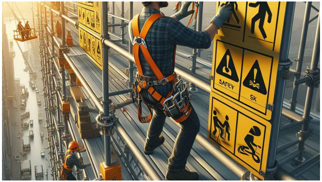
Disclaimer: we are a broad group who needs to get to know each others scientifically.
Apologies if I don't "hit" the right level for all.
I have favored simplicity over "absolute" rigor.

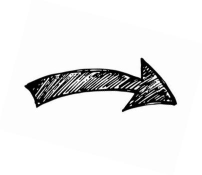
Outline

Formally Defining Safety?

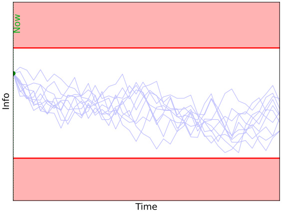


Safety in the real world





Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

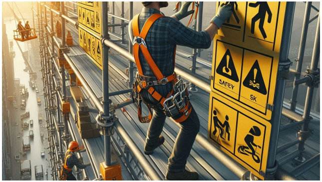
Performance

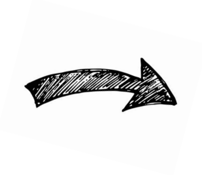
Utility

Action

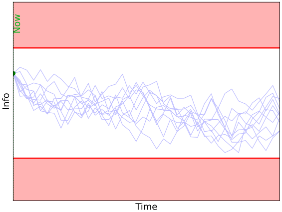
Information

Safety in the real world

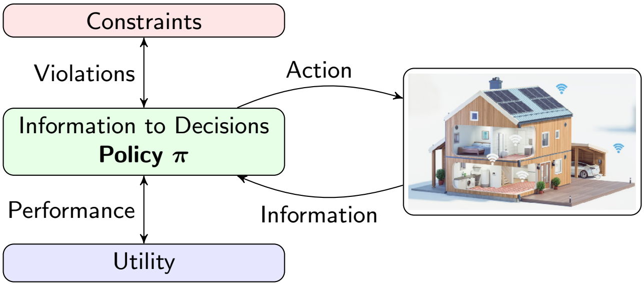




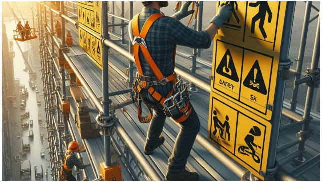
Safety in the math world



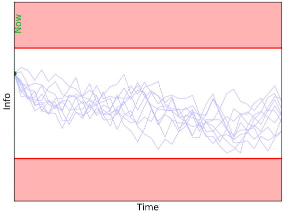
Formally Defining Safety?



Safety in the real world



Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

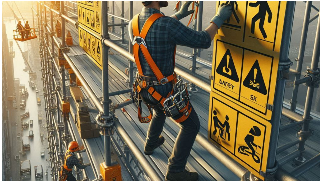
Action

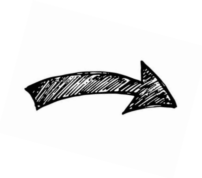
Information

Performance

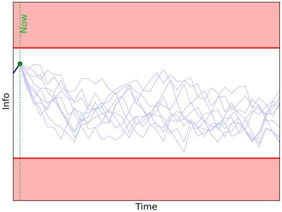
Utility

Safety in the real world





Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

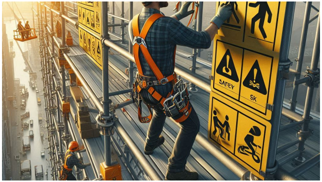
Performance

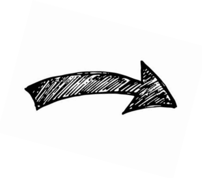
Utility

Action

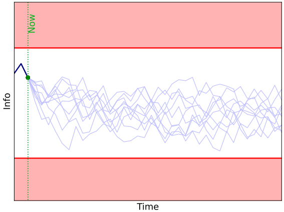
Information

Safety in the real world





Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

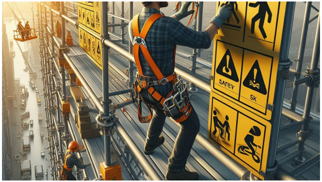
Action

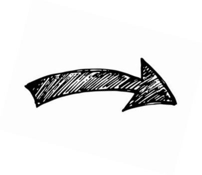
Information

Performance

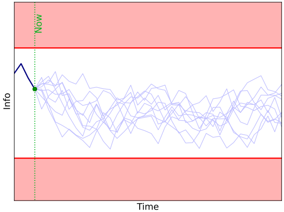
Utility

Safety in the real world





Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

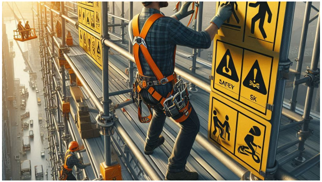
Performance

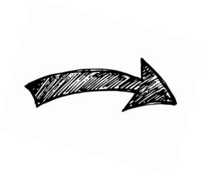
Utility

Action

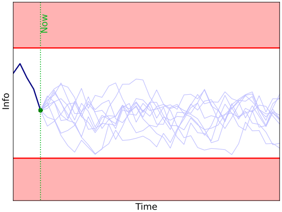
Information

Safety in the real world





Safety in the math world



Formally Defining Safety?

Constraints

Violations

Information to Decisions
Policy π

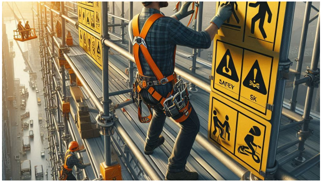
Action

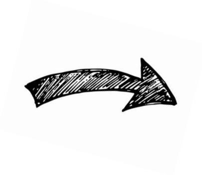
Information

Performance

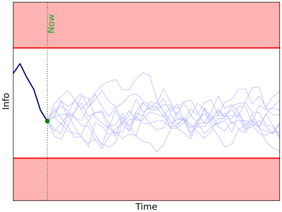
Utility

Safety in the real world

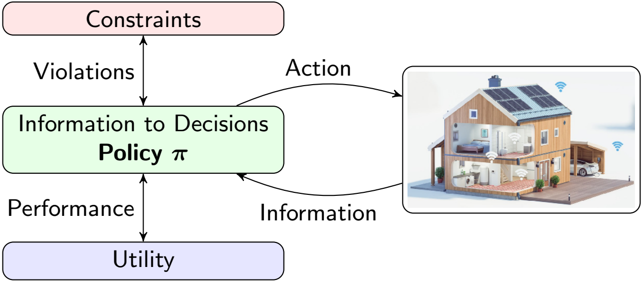




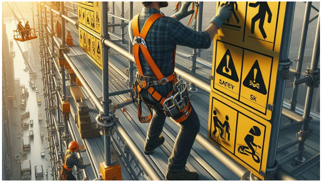
Safety in the math world

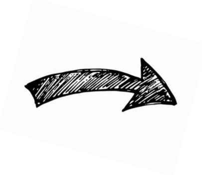


Formally Defining Safety?

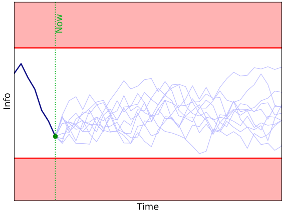


Safety in the real world

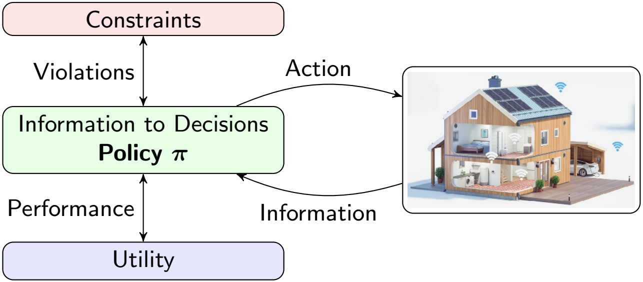




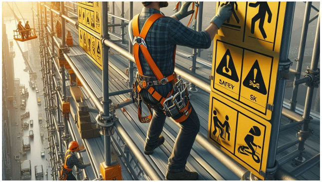
Safety in the math world

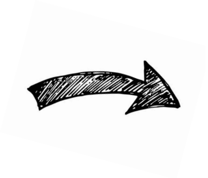


Formally Defining Safety?

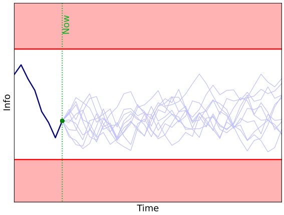


Safety in the real world





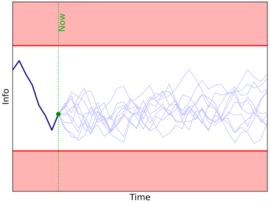
Safety in the math world



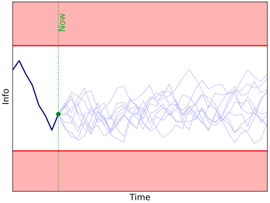
Constrained MDPs

In words

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$



Constrained MDPs



In words

$$\pi^\star = \underset{\pi}{\arg\max} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$

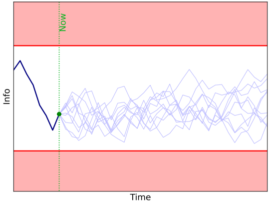
Formally

$$\pi^\star = \underset{\pi}{\arg\max} \quad \mathrm{E}\left[\sum_{\text{time}} \left(\text{Utility} - \begin{cases} 0 & \text{if} \quad \text{Constraint ok} \\ \infty & \text{if} \quad \text{Constraint not ok} \end{cases}\right)\right]$$

Constrained MDPs

In words

$$\boldsymbol{\pi}^\star = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$



Formally

$$\boldsymbol{\pi}^\star = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathrm{E}\left[\sum_{\text{time}} \left(\text{Utility} - \begin{cases} 0 & \text{if} \quad \text{Constraint ok} \\ \infty & \text{if} \quad \text{Constraint not ok} \end{cases}\right)\right]$$

Build policy using

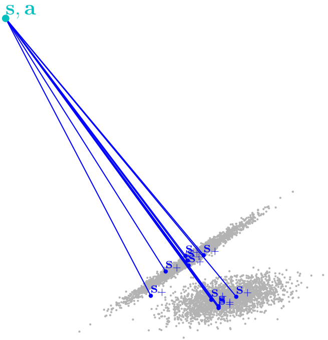
- Perfect model of the real world $\hat{\mathrm{P}}[\mathbf{s}_+|\mathbf{s},\mathbf{a}] = \mathrm{P}[\mathbf{s}_+|\mathbf{s},\mathbf{a}]$
- Model "pessimistic" about the uncertainties

... to evaluate "$\mathrm{E}[\cdot]$"

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Distribution of one-step forward

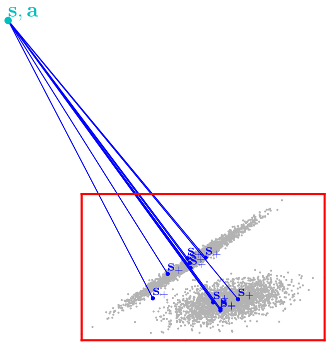


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Distribution of one-step forward with "container"

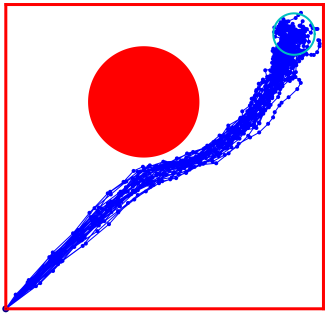


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories

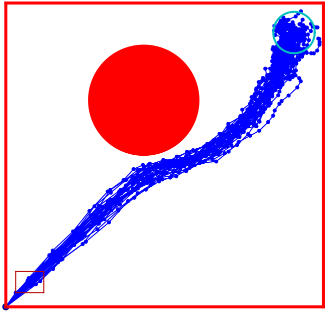


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

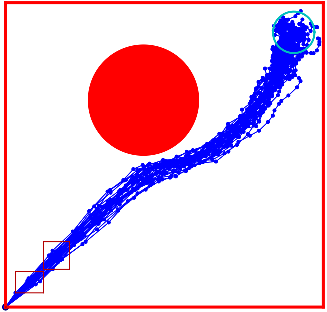


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

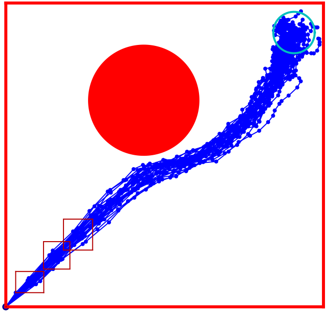


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

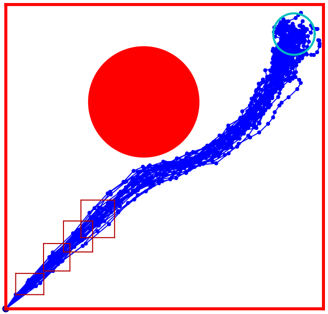


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

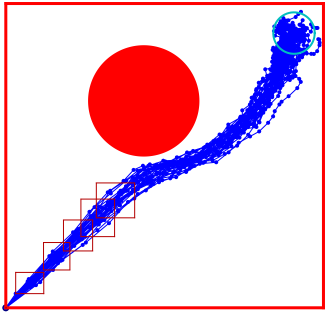


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

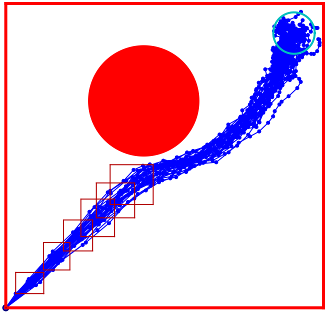


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

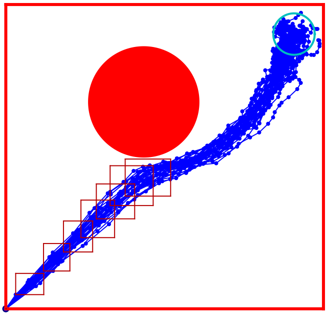


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

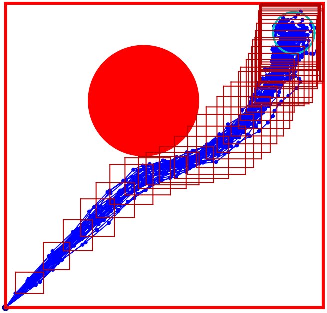


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"

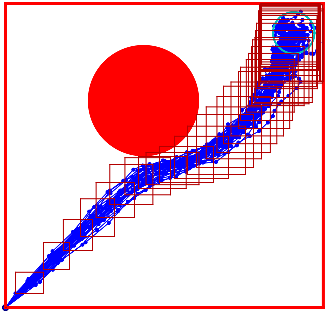


Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

Pessimistic Models for Decision Making

- Model must "contain" the uncertainty
- "Container" (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will "cover" the real world
- Decision policy wants to be safe w.r.t. the "containers"

Trajectories with "containers"



Remarks

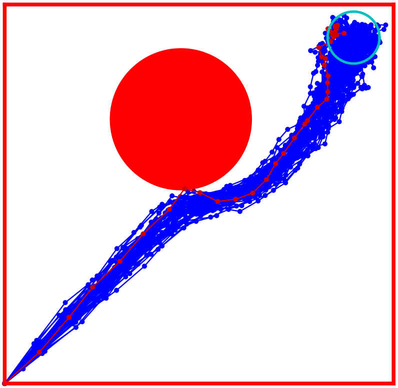
- The propagation of the "containers" in the model predictions can be expensive / difficult
- Pessimistic propagations are usually needed → pessimistic over pessimistic
- Policy based on worst-case perspective makes the decisions highly conservative
- Often labelled "Robust" decision making

Illustrate: wide & thin-tailed distribution requires a big box, to contain transitions that almost surely won't happen. SG can do on 11.9

MDPs with probabilistic safety

In words

$$\begin{aligned} \boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad & \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right] \\ \text{s.t.} \quad & \text{Probability of no violation} \geq c \end{aligned}$$



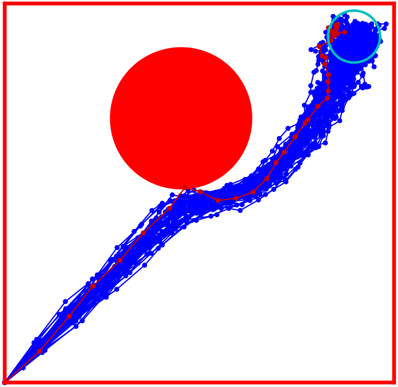
MDPs with probabilistic safety

In words

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Probability of no violation} \geq c$$

Formally

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right]$$
$$\text{s.t.} \quad \mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right] \geq c$$



MDPs with probabilistic safety

In words

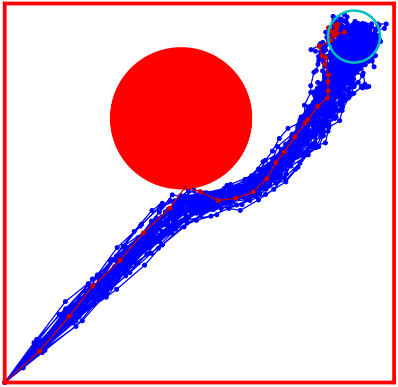
$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Probability of no violation} \geq c$$

Formally

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right]$$
$$\text{s.t.} \quad \mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right] \geq c$$

Remarks:

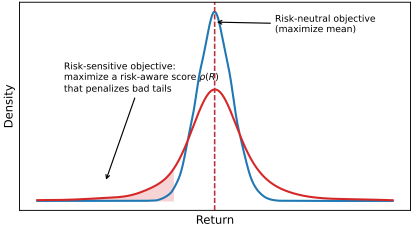
- If we can tolerate $c > 0$ (small) it can make a big gain in performance
- Aligned with industrial / practical standards on "large series"
- Problem needs a "termination" (time or goal reached)
- Building decisions can be difficult from a computational point of view



Risk-sensitive MDPs

In words

- Risk-sensitive MDPs optimizes a **risk measure** to account for the severity of rare outcomes.
- Intuition: a "tail-risk filter" even if outcomes are good most of the time, severe tails influence decisions.



SG: nice slide. Good to stress that risk-measures are looking at risks through the utility function, as opposed to through undesirable events (not necessarily related to utility). This is an important difference not obvious in the slide. In addition, can we plot an example of ρ and comment on what it does? Also relate ρ to expected value?

Mean-optimal can ignore tails; risk-sensitive downweights heavy losses.

Risk-sensitive MDPs

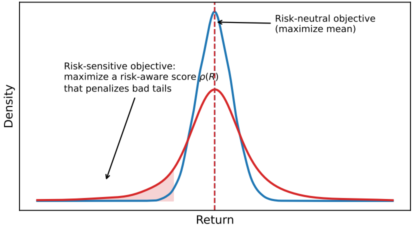
In words

- Risk-sensitive MDPs optimizes a **risk measure** to account for the severity of rare outcomes.
- Intuition: a "tail-risk filter" even if outcomes are good most of the time, severe tails influence decisions.

Formally

$$\pi^* = \arg\max_{\pi} \rho\Big(\sum_{t=0}^{\infty} \gamma^t r_t\Big)$$

ρ is a risk measure (e.g., VaR, Entropic, CVaR).



SG: nice slide. Good to stress that risk-measures are looking at risks through the utility function, as opposed to through undesirable events (not necessarily related to utility). This is an important difference not obvious in the slide. In addition, can we plot an example of ρ and comment on what it does? Also relate ρ to expected value?

Mean-optimal can ignore tails; risk-sensitive downweights heavy losses.

Risk-sensitive MDPs

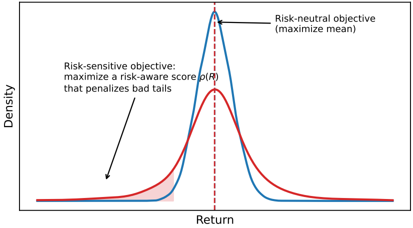
In words

- Risk-sensitive MDPs optimizes a **risk measure** to account for the severity of rare outcomes.
- Intuition: a "tail-risk filter" even if outcomes are good most of the time, severe tails influence decisions.

Formally

$$\pi^* = \arg\max_{\pi} \rho\Big(\sum_{t=0}^{\infty} \gamma^t r_t\Big)$$

ρ is a risk measure (e.g., VaR, Entropic, CVaR).



Mean-optimal can ignore tails; risk-sensitive downweights heavy losses.

SG: nice slide. Good to stress that risk-measures are looking at risks through the utility function, as opposed to through undesirable events (not necessarily related to utility). This is an important difference not obvious in the slide. In addition, can we plot an example of ρ and comment on what it does? Also relate ρ to expected value?

Remarks:

- Often possile to solve using classical methods such as DP.
- Risk-sensitive criteria do *not* guarantee a bound on violation.

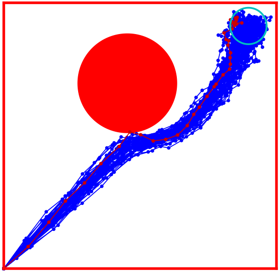
Outline

Robust Repeated Planning

- Introduction: Robust MPC, Scenario Trees, MC, robust multi-stage stochastic programing, MPPI?
- Difficulties: guarantees for nonlinear systems, persistent safety (recursive feasibility)

Dynamic Programming & RL with Probabilistic Safety

$$\begin{aligned} \boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad & \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right] \\ \text{s.t.} \quad & \mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right] \geq c \end{aligned}$$



Expected value form

$$\begin{aligned} \boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad & \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k, \mathbf{a}_k)\right] \\ \text{s.t.} \quad & \mathrm{E}\left[\begin{array}{ll} 1 & \text{if no violation occurred} \\ 0 & \text{otherwise} \end{array}\right] \geq c \end{aligned}$$

- Expected value form enables classical techniques (ref. 1st lecture), i.e. DP and RL
- Difficulties: estimate expected values (sample based) when $\mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right]$ is close to 1. Illustrate?? SG will try on 11.9

Outline

Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run $\rightarrow$ "Safe set" $\mathrm{S}(\mathrm{s})$

Safety Filters & Reinforcement Learning

- For any state s, an “oracle” tells you which actions will not jeopardize safety in the long run$\rightarrow$“Safe set” $\mathrm{S}\left(\mathrm{s}\right)$
- We can learn an “unsafe” policy

$$\boldsymbol{\pi}^{\star} = \operatorname*{arg\,max}_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

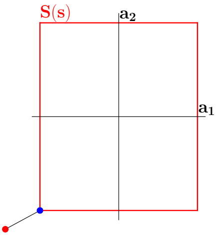
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$



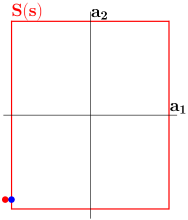
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathbb{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$



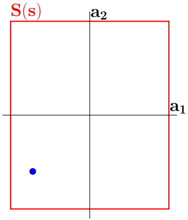
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\arg\max} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: Action $=$ Projection$_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$



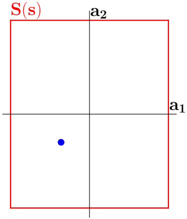
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$



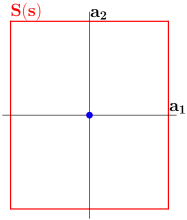
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}(\mathbf{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathbf{s})}\left[\boldsymbol{\pi}^\star(\mathbf{s})\right]$



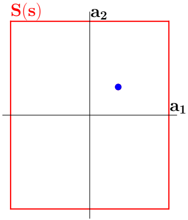
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}\left(\mathbf{s}\right)$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: Action $=$ Projection$_{\mathrm{S}(\mathbf{s})}\left[\boldsymbol{\pi}^{\star}\left(\mathbf{s}\right)\right]$



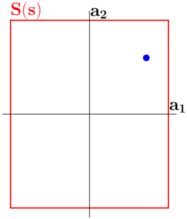
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\arg\max} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$



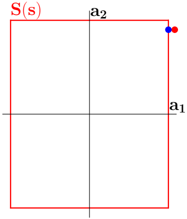
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}\left(\mathrm{s}\right)$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: Action $=$ Projection$_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^\star\left(\mathrm{s}\right)\right]$



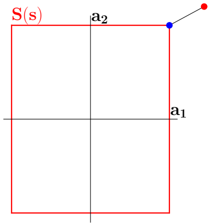
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run $\rightarrow$ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathbf{s})}\left[\boldsymbol{\pi}^{\star}(\mathbf{s})\right]$



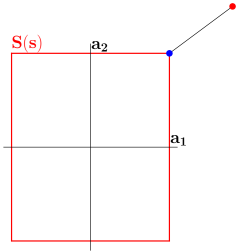
Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathbb{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^\star(\mathrm{s})\right]$



Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

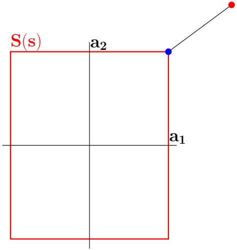
$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$

Remark: "oracle" can be very hard to build, or not...

Not losing the king



Safety Filters & Reinforcement Learning

- For any state s, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$ "Safe set" $\mathbb{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathbb{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

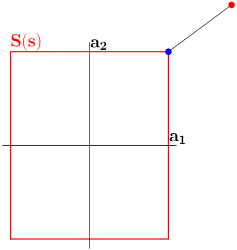
that focuses only on utility

- For state s, take the decisions using: $\text{Action} = \text{Projection}_{\mathbb{S}(\mathrm{s})}\left[\boldsymbol{\pi}^{\star}(\mathrm{s})\right]$

Remark: "oracle" can be very hard to build, or not...

Not losing the king

Able to stop within the visible distance



Safety Filters & Reinforcement Learning

- For any state $\mathbf{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run→ "Safe set" $\mathbb{S}(\mathbf{s})$
- We can learn an "unsafe" policy

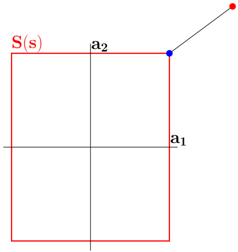
$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathbb{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state $\mathbf{s}$, take the decisions using: Action $= \text{Projection}_{\mathbb{S}(\mathbf{s})}\left[\boldsymbol{\pi}^\star(\mathbf{s})\right]$

Remarks:

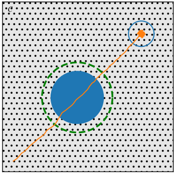
- Often close to practice, e.g., flight envelop protection in modern airplane, semi-autonomous driving in cars, etc
- Oracle ⇒ knowledge-based & conservative!



Control Barrier Functions (CBFs)

In words

- CBFs provide a formal, model-based way to build the "oracle" safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it be safe.



Unsafe trajectory: violates $h(x) \geq 0$.

Control Barrier Functions (CBFs)

In words

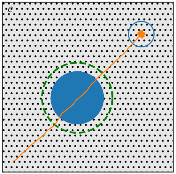
- CBFs provide a formal, model-based way to build the "oracle" safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it be safe.

Formally

$$\dot{x} = f(x) + g(x)u, \qquad \mathcal{C} = \{x : h(x) \geq 0\}$$

$$u^\star(x) = \arg\min_u \tfrac{1}{2}\|u - u_{\mathrm{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \geq 0}_{\text{CBF constraint}}$$



Unsafe trajectory: violates $h(x) \geq 0$.

Control Barrier Functions (CBFs)

In words

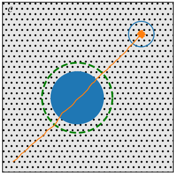
- CBFs provide a formal, model-based way to build the "oracle" safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it be safe.

Formally

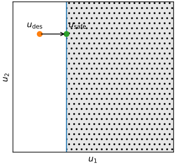
$$\dot{x} = f(x) + g(x)u, \qquad \mathcal{C} = \{x : h(x) \geq 0\}$$

$$u^\star(x) = \arg\min_u \tfrac{1}{2}\|u - u_{\mathrm{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \geq 0}_{\text{CBF constraint}}$$



Unsafe trajectory: violates $h(x) \geq 0$.



CBF filter: project u_{des} to feasible u_{safe}.

Control Barrier Functions (CBFs)

In words

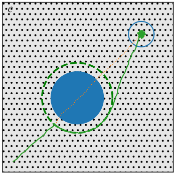
- CBFs provide a formal, model-based way to build the "oracle" safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it be safe.

Formally

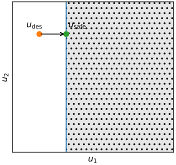
$$\dot{x} = f(x) + g(x)u, \qquad \mathcal{C} = \{x : h(x) \geq 0\}$$

$$u^\star(x) = \arg\min_u \tfrac{1}{2}\|u - u_{\mathrm{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \geq 0}_{\text{CBF constraint}}$$



CBF-filtered trajectory.

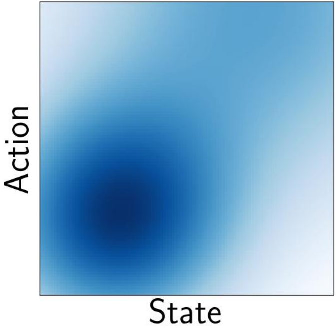


CBF filter: project u_{des} to feasible u_{safe}.

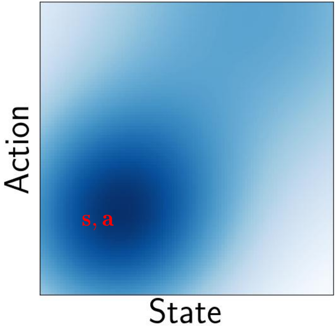
- The barrier function and safe set $\mathcal{C}$ are typically **constructed from domain knowledge** (physics, rules, safety envelopes).
- Requires a (possibly simplified, conservative) **system model**: **robust** approach.

Outline

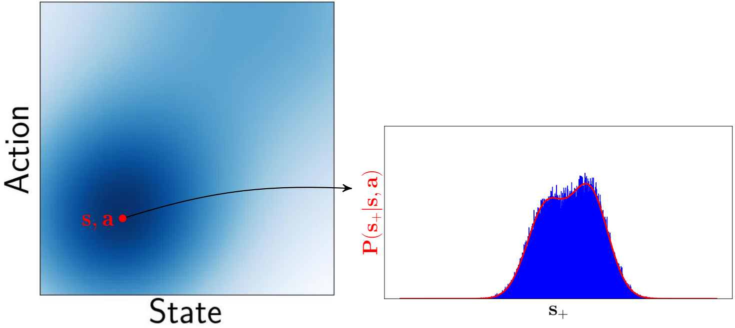
Data & Epistemic Uncertainty



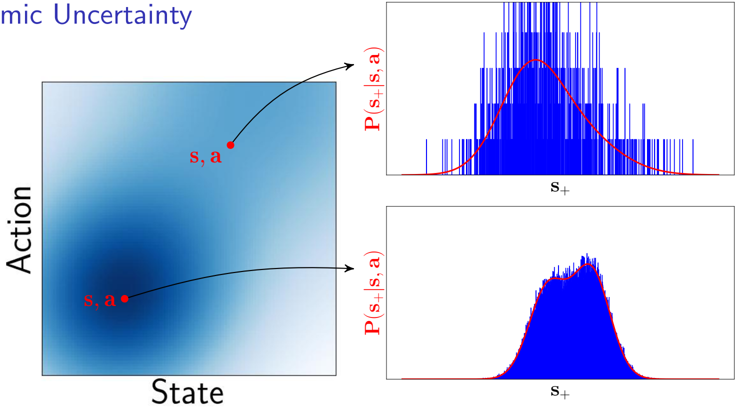
Data & Epistemic Uncertainty



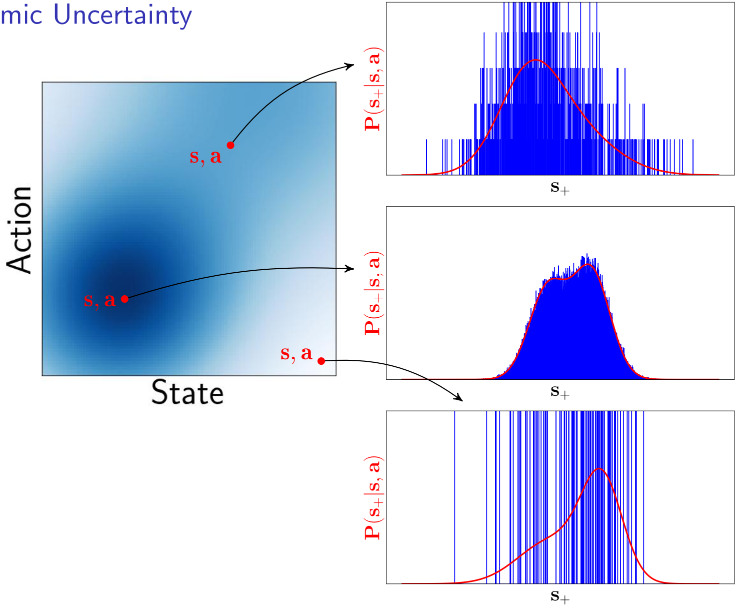
Data & Epistemic Uncertainty



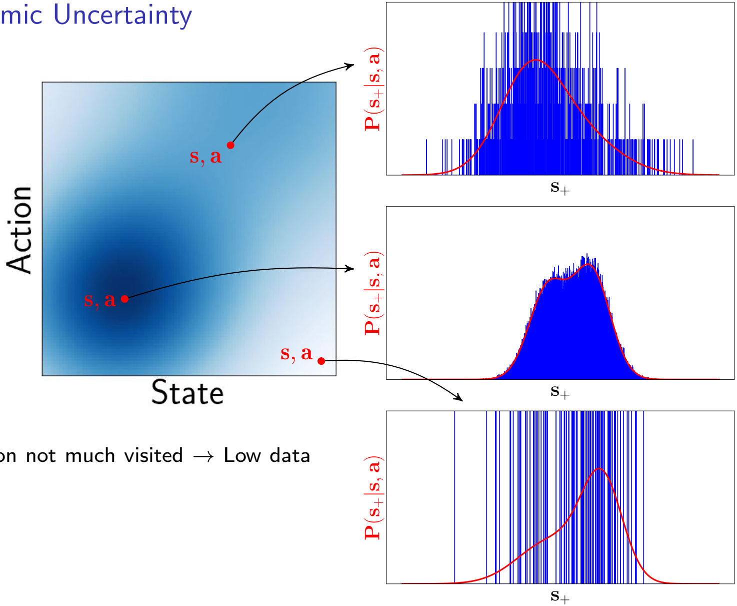
Data & Epistemic Uncertainty



Data & Epistemic Uncertainty



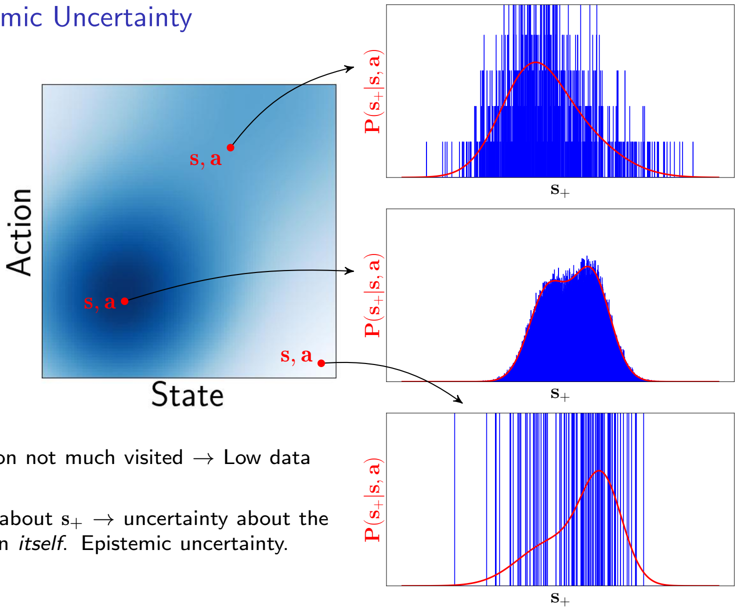
Data & Epistemic Uncertainty



Remarks

- State-action not much visited → Low data about s_+

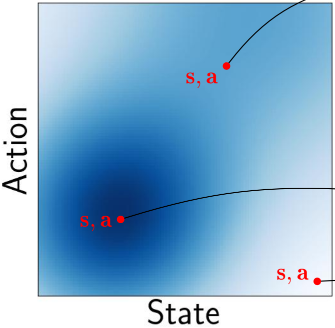
Data & Epistemic Uncertainty

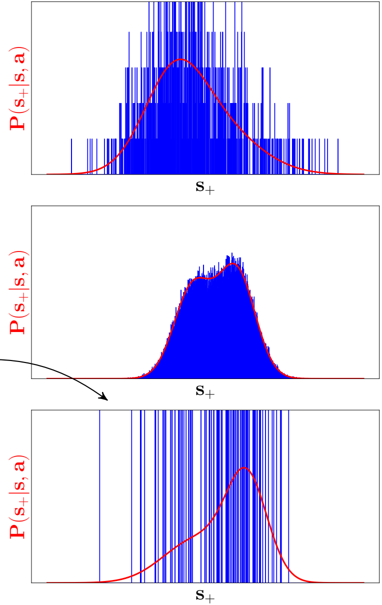


Remarks

- State-action not much visited $\rightarrow$ Low data about s_+
- Low data about s_+ $\rightarrow$ uncertainty about the distribution *itself*. Epistemic uncertainty.

Data & Epistemic Uncertainty

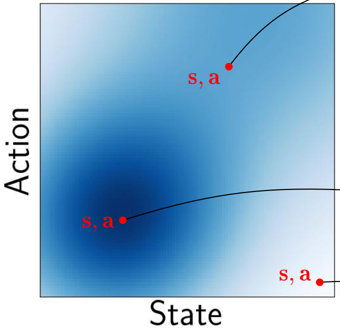


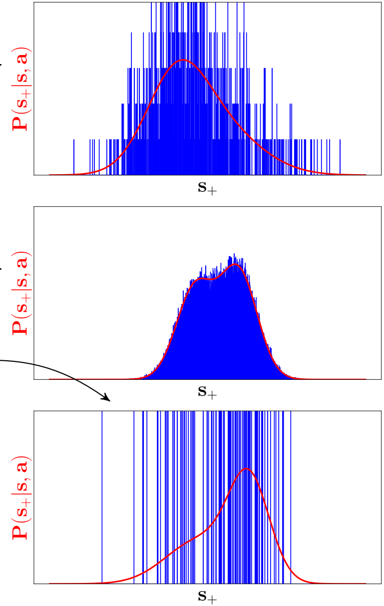


Remarks

- State-action not much visited → Low data about s_+
- Low data about s_+ → uncertainty about the distribution *itself*. Epistemic uncertainty.
- Continuous state-action problems → single data point per $\mathrm{s}, \mathbf{a}$

Data & Epistemic Uncertainty





Remarks

- State-action not much visited $\rightarrow$ Low data about s_+
- Low data about s_+ $\rightarrow$ uncertainty about the distribution *itself*. Epistemic uncertainty.
- Continuous state-action problems $\rightarrow$ single data point per $\mathrm{s}, \mathbf{a}$
- Rely on regularity across $\mathrm{s}, \mathbf{a}$!!

Distributionally Robust Methods

AKHIL?

Out-Of-Distribution (OOD) “Guard” Methods

SG can do on 11.9