Bayes


RL

Gaussian Process Temporal Difference Learning

Yaakov Engel

Collaborators: Shie Mannor, Ron Meir

ALBERTA INGENUITY CENTRE FOR MACHINE LEARNING

UNIVERSITY OF ALBERTA

Bayes
RL

Why use GPs in RL?

- A Bayesian approach to value estimation
- Forces us to to make our assumptions explicit
- Non-parametric – priors are placed and inference is performed directly in function space (kernels).
- But, can also be defined parametrically
- Domain knowledge intuitively coded in priors
- Provides full posterior over values, not just point estimates
- Efficient, on-line implementations, suitable for large problems

Gaussian Processes

Definition: "An **indexed** set of jointly Gaussian random variables"

Note: The index set $\mathcal{X}$ may be just about **any** set.

Example: $F(\boldsymbol{x})$, index is $\boldsymbol{x} \in [0,1]^n$

F's distribution is specified by its mean and covariance:

$$\mathbf{E}\big[F(\boldsymbol{x})\big] = m(\boldsymbol{x}) \ , \quad \mathbf{Cov}\big[F(\boldsymbol{x}), F(\boldsymbol{x}')\big] = k(\boldsymbol{x}, \boldsymbol{x}')$$

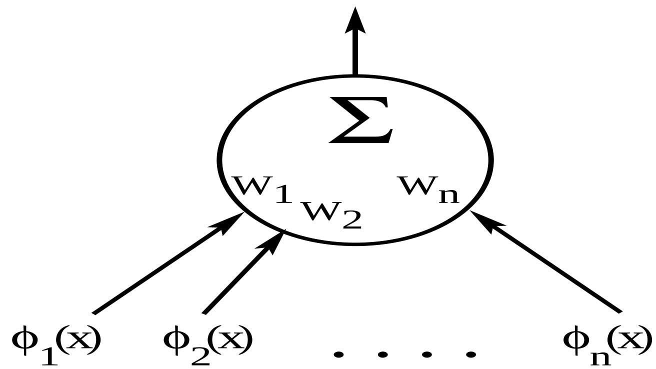
Conditions on k:

Symmetric, positive definite $\Rightarrow$ k is a **Mercer kernel**

Example: Parametric GP

A linear combination of basis functions:

$$\boxed{F(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^\top W}$$



If $W \sim \mathcal{N}\{\mathbf{m_w}, \mathbf{C_w}\}$,

then F **is a GP:**

$$\mathbf{E}[F(\boldsymbol{x})] = \boldsymbol{\phi}(\boldsymbol{x})^\top \mathbf{m_w},$$

$$\mathbf{Cov}[F(\boldsymbol{x}), F(\boldsymbol{x}')] = \boldsymbol{\phi}(\boldsymbol{x})^\top \mathbf{C_w} \boldsymbol{\phi}(\boldsymbol{x}')$$

Conditioning – Gauss-Markov Thm.

Theorem *Let Z and Y be random vectors jointly distributed according to the multivariate normal distribution*

$$\begin{pmatrix} Z \\ Y \end{pmatrix} \sim \mathcal{N}\left\{ \begin{pmatrix} \mathbf{m}_z \\ \mathbf{m}_\mathbf{y} \end{pmatrix}, \begin{bmatrix} \mathbf{C}_{zz} & \mathbf{C}_{z\mathbf{y}} \\ \mathbf{C}_{\mathbf{y}z} & \mathbf{C}_{\mathbf{yy}} \end{bmatrix} \right\}.$$

Then $Z|Y \sim \mathcal{N}\left\{\hat{Z}, \mathbf{P}\right\}$, *where*

$$\boxed{\begin{aligned} \hat{Z} &= \mathbf{m}_z + \mathbf{C}_{z\mathbf{y}}\mathbf{C}_{\mathbf{yy}}^{-1}(Y - \mathbf{m}_\mathbf{y}) \\ \mathbf{P} &= \mathbf{C}_{zz} - \mathbf{C}_{z\mathbf{y}}\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{C}_{\mathbf{y}z}. \end{aligned}}$$

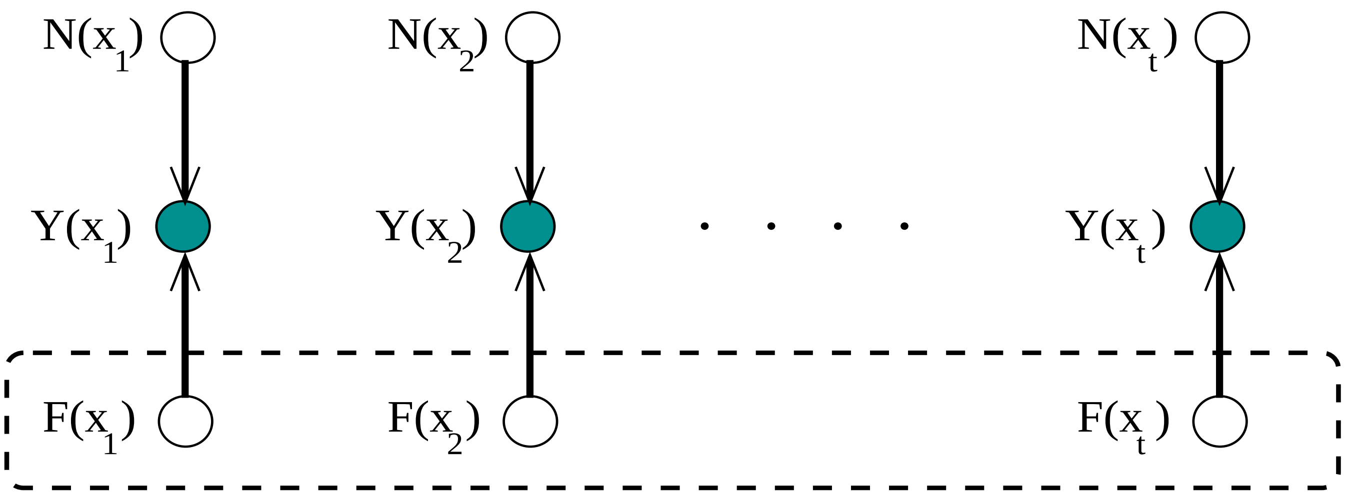
Bayes RL

GP Regression

Sample: $((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t))$

Model equation: $\boxed{Y(\boldsymbol{x}_i) = F(\boldsymbol{x}_i) + N(\boldsymbol{x}_i)}$

GP Prior on F: $F \sim \mathcal{N}\left\{0, k(\cdot,\cdot)\right\}$



N**:** IID zero-mean Gaussian noise, with variance σ^2

Bayes RL

GP Regression (ctd.)

Denote:

$$Y_t = \left(Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_t)\right)^\top,$$

$$\mathbf{k}_t(\boldsymbol{x}) = \left(k(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_t, \boldsymbol{x})\right)^\top,$$

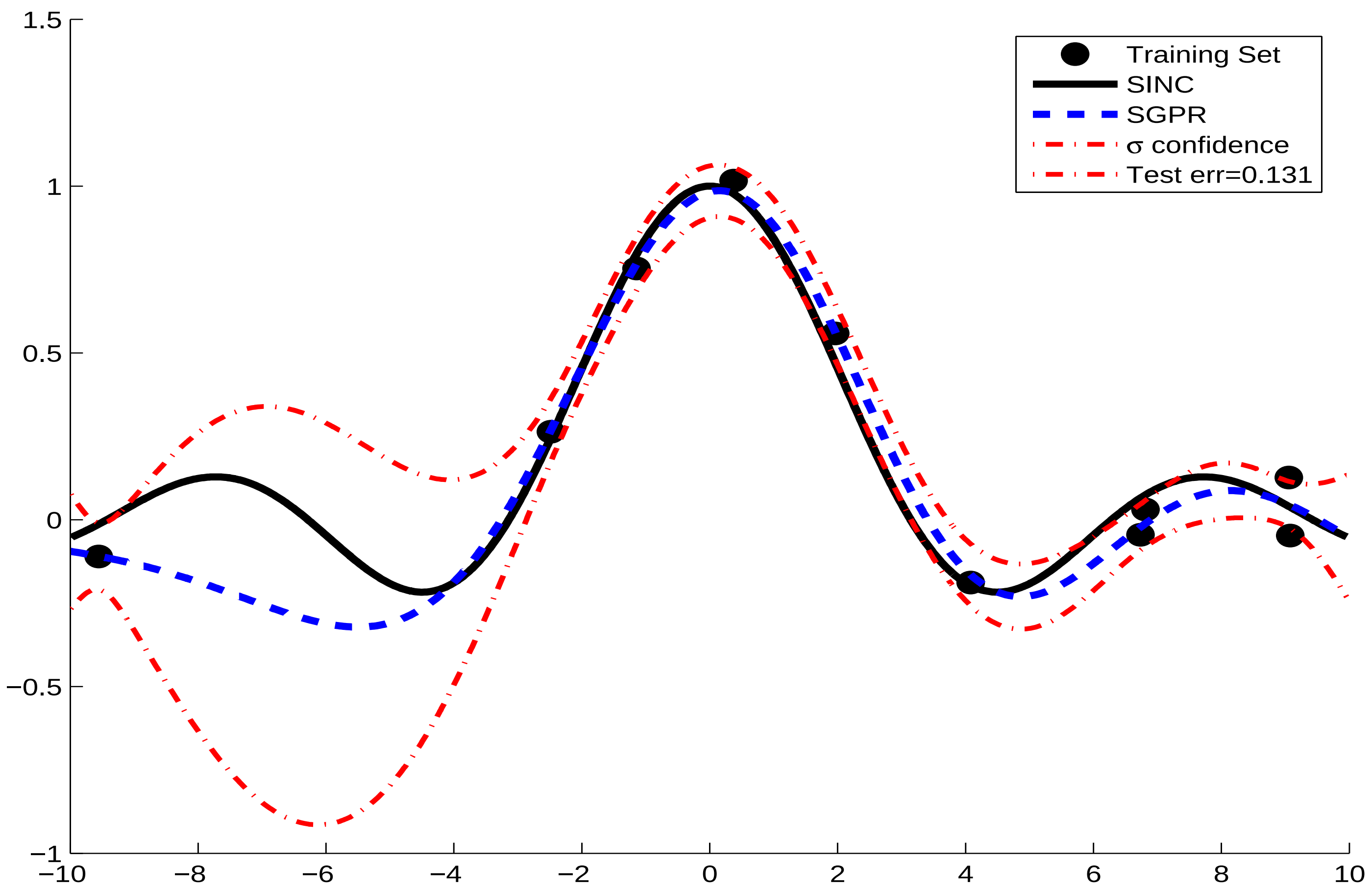
$$\mathbf{K}_t = \left[\mathbf{k}_t(\boldsymbol{x}_1), \ldots, \mathbf{k}_t(\boldsymbol{x}_t)\right].$$

Then:

$$\begin{pmatrix} F(\boldsymbol{x}) \\ Y_t \end{pmatrix} \sim \mathcal{N}\left\{ \begin{pmatrix} 0 \\ \mathbf{0} \end{pmatrix}, \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}) & \mathbf{k}_t(\boldsymbol{x})^\top \\ \mathbf{k}_t(\boldsymbol{x}) & \mathbf{K}_t + \sigma^2 \mathbf{I} \end{bmatrix} \right\}$$

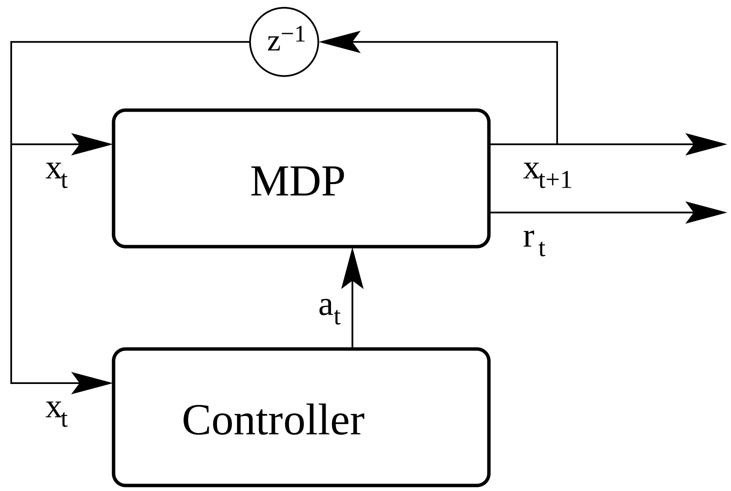
Now apply conditioning formula to compute the posterior moments of $F(\boldsymbol{x})$, given $Y_t = \mathbf{y}_t = (y_1, \ldots, y_t)^\top$.

EXAMPLE



1.5
1
0.5
0
−0.5
−1
−10
−8
−6
−4
−2
0
2
4
6
8
10
Training Set
SINC
SGPR
σ confidence
Test err=0.131

Markov Decision Processes



State space: $\mathcal{X}$, state $\boldsymbol{x} \in \mathcal{X}$

Action space: $\mathcal{A}$, action $\boldsymbol{a} \in \mathcal{A}$

Joint state-action space: $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$, $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{a})$

Transition prob. density: $\boldsymbol{x}_{t+1} \sim p(\cdot|\boldsymbol{x}_t, \boldsymbol{a}_t)$

Reward prob. density: $R(\boldsymbol{x}_t, \boldsymbol{a}_t) \sim q(\cdot|\boldsymbol{x}_t, \boldsymbol{a}_t)$

Control and Returns

Stationary policy: $\boldsymbol{a}_t \sim \mu(\cdot|\boldsymbol{x}_t)$

Path: $\boldsymbol{\xi}^\mu = (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots)$

Discounted Return: $D(\boldsymbol{\xi}^\mu) = \sum_{i=0}^{\infty} \gamma^i R(\boldsymbol{z}_i)$

Value function: $V^\mu(\boldsymbol{x}) = \mathbf{E}_\mu[D(\boldsymbol{\xi}^\mu)|\boldsymbol{x}_0 = \boldsymbol{x}]$

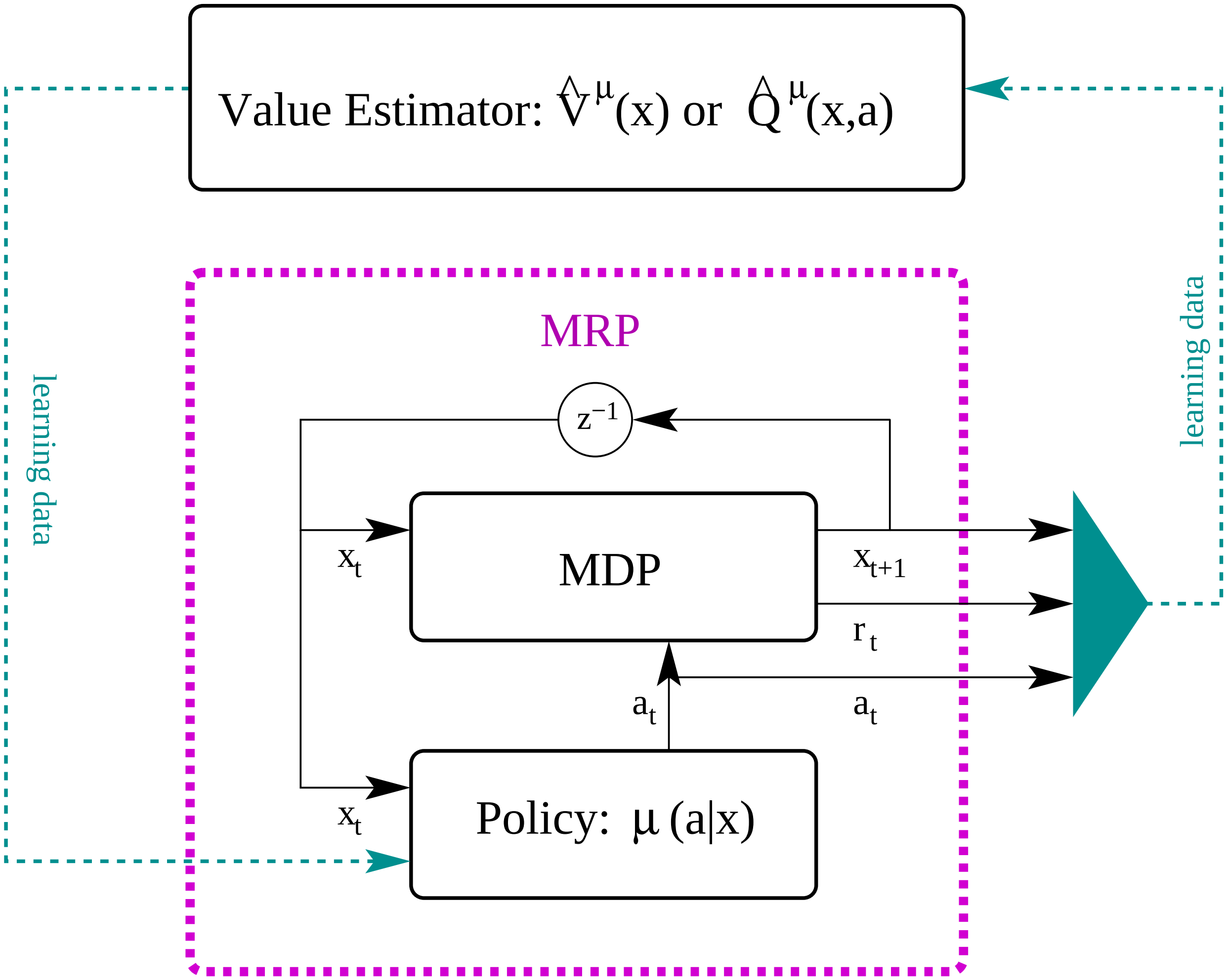
State-action value func.: $Q^\mu(\boldsymbol{z}) = \mathbf{E}_\mu[D(\boldsymbol{\xi}^\mu)|\boldsymbol{z}_0 = \boldsymbol{z}]$

Goal: Find a policy μ^* maximizing $V^\mu(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{X}$

Note: If $Q^*(\boldsymbol{x}, \boldsymbol{a}) = Q^{\mu^*}(\boldsymbol{x}, \boldsymbol{a})$ is available, then an optimal action for state $\boldsymbol{x}$ is given by any $\boldsymbol{a}^* \in \operatorname{argmax}_{\boldsymbol{a}} Q^*(\boldsymbol{x}, \boldsymbol{a})$.

Bayes RL

Value-Based RL



Bellman's Equation

For a fixed policy μ:

$$V^{\mu}(\boldsymbol{x}) = \mathbf{E}_{\boldsymbol{x}',\boldsymbol{a}|\boldsymbol{x}}\left[\bar{R}(\boldsymbol{x},\boldsymbol{a}) + \gamma V^{\mu}(\boldsymbol{x}')\right]$$

Optimal value and policy:

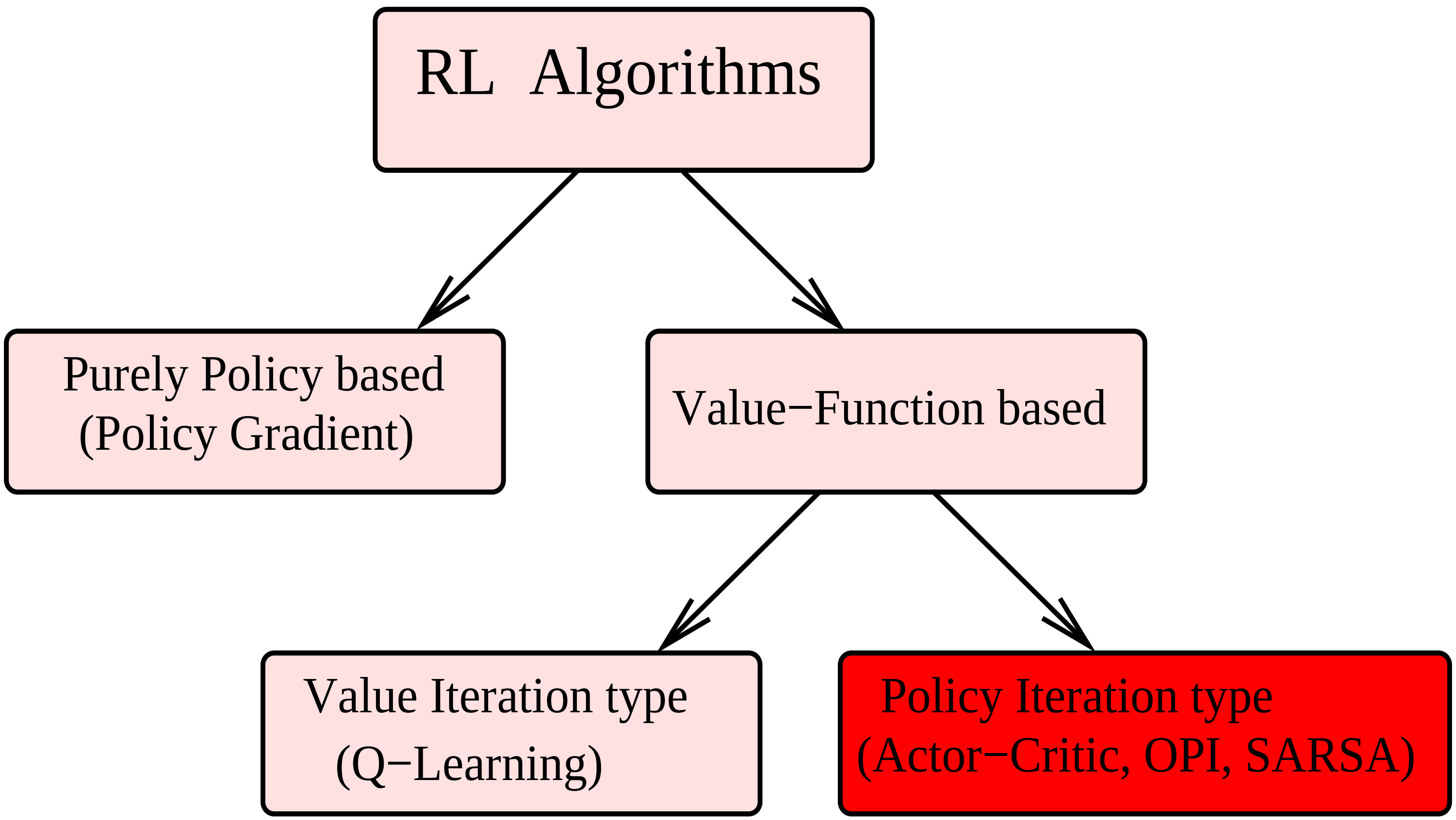
$$V^{*}(\boldsymbol{x}) = \max_{\mu} V^{\mu}(\boldsymbol{x}) \; , \quad \mu^{*} = \operatorname*{argmax}_{\mu} V^{\mu}(\boldsymbol{x})$$

How to solve it?

- Methods based on Value Iteration (e.g. Q-learning)
- Methods based on Policy Iteration (e.g. SARSA, OPI, Actor-Critic)

Bayes RL

SOLUTION METHOD TAXONOMY



PI methods need a "subroutine" for policy evaluation

Bayes

RL

What's Missing?

Shortcomings of current policy evaluation methods:

- Some methods can only be applied to small problems
- No probabilistic interpretation - how good is the estimate?
- Only parametric methods are capable of operating on-line
- Non-parametric methods are more flexible but only work off-line
- Small-step-size (stoch. approx.) methods use data inefficiently
- Finite-time solutions lack interpretability, all statements are asymptotic
- Convergence issues

GP Temporal Difference Learning

Model Equations:

$$R(\boldsymbol{x}_i) = V(\boldsymbol{x}_i) - \gamma V(\boldsymbol{x}_{i+1}) + N(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$$

Or, in compact form:

$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t$$

$$\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ 0 & 1 & -\gamma & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -\gamma \end{bmatrix}.$$

Our (Bayesian) goal:

Find the posterior distribution of V,
given a sequence of observed states and rewards.

Bayes

RL

Deterministic Dynamics

Bellman's Equation:

$$V(\boldsymbol{x}_i) = \bar{R}(\boldsymbol{x}_i) + \gamma V(\boldsymbol{x}_{i+1})$$

Define:

$$N(\boldsymbol{x}) = R(\boldsymbol{x}) - \bar{R}(\boldsymbol{x})$$

Assumption: $N(\boldsymbol{x}_i)$ are Normal, IID, with variance σ^2.

Model Equations:

$$\boxed{R(\boldsymbol{x}_i) = V(\boldsymbol{x}_i) - \gamma V(\boldsymbol{x}_{i+1}) + N(\boldsymbol{x}_i)}$$

In compact form:

$$\boxed{R_t = \mathbf{H}_{t+1} V_{t+1} + N_t \text{ , with } N_t \sim \mathcal{N}\left\{0, \sigma^2 \mathbf{I}\right\}}$$

Bayes RL

Stochastic Dynamics

The discounted return:

$$D(\boldsymbol{x}_i) = \mathbf{E}_\mu D(\boldsymbol{x}_i) + (D(\boldsymbol{x}_i) - \mathbf{E}_\mu D(\boldsymbol{x}_i)) = V(\boldsymbol{x}_i) + \Delta V(\boldsymbol{x}_i)$$

For a stationary MDP:

$$D(\boldsymbol{x}_i) = R(\boldsymbol{x}_i) + \gamma D(\boldsymbol{x}_{i+1}) \text{ (where } \boldsymbol{x}_{i+1} \sim p(\cdot|\boldsymbol{x}_i, \boldsymbol{a}_i),\ \boldsymbol{a}_i \sim \mu(\cdot|\boldsymbol{x}_i))$$

Substitute and rearrange:

$$R(\boldsymbol{x}_i) = V(\boldsymbol{x}_i) - \gamma V(\boldsymbol{x}_{i+1}) + N(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$$
$$N(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \stackrel{\text{def}}{=} \Delta V(\boldsymbol{x}_i) - \gamma \Delta V(\boldsymbol{x}_{i+1})$$

Assumption: $\Delta V(\boldsymbol{x}_i)$ are Normal, i.i.d., with variance σ^2.

In compact form:

$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t \text{ , with } N_t \sim \mathcal{N}\left\{0, \sigma^2 \mathbf{H}_{t+1} \mathbf{H}_{t+1}^\top\right\}$$

The Posterior

General noise covariance: $\boxed{\mathbf{Cov}[N_t] = \mathbf{\Sigma}_t}$

Joint distribution:

$$\begin{bmatrix} R_{t-1} \\ V(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left\{ \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^\top + \mathbf{\Sigma}_t & \mathbf{H}_t \mathbf{k}_t(\boldsymbol{x}) \\ \mathbf{k}_t(\boldsymbol{x})^\top \mathbf{H}_t^\top & k(\boldsymbol{x}, \boldsymbol{x}) \end{bmatrix} \right\}$$

Condition on R_{t-1}:

$$\mathbf{E}[V(\boldsymbol{x}) | R_{t-1} = \mathbf{r}_{t-1}] = \mathbf{k}_t(\boldsymbol{x})^\top \boldsymbol{\alpha}_t$$

$$\mathbf{Cov}[V(\boldsymbol{x}), V(\boldsymbol{x}') | R_{t-1} = \mathbf{r}_{t-1}] = k(\boldsymbol{x}, \boldsymbol{x}') - \mathbf{k}_t(\boldsymbol{x})^\top \mathbf{C}_t \mathbf{k}_t(\boldsymbol{x}')$$

$$\boldsymbol{\alpha}_t = \mathbf{H}_t^\top \left(\mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^\top + \mathbf{\Sigma}_t \right)^{-1} \mathbf{r}_{t-1}, \quad \mathbf{C}_t = \mathbf{H}_t^\top \left(\mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^\top + \mathbf{\Sigma}_t \right)^{-1} \mathbf{H}_t.$$

Bayes

RL

Learning State-Action Values

Under a fixed stationary policy μ, state-action pairs $\boldsymbol{z}_t$ form a Markov chain, just like the states $\boldsymbol{x}_t$.

Consequently $Q^\mu(\boldsymbol{z})$ behaves similarly to $V^\mu(\boldsymbol{x})$:

$$R(\boldsymbol{z}_i) = Q(\boldsymbol{z}_i) - \gamma Q(\boldsymbol{z}_{i+1}) + N(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$$

Posterior moments:

$$\mathbf{E}[Q(\boldsymbol{z}) | R_{t-1} = \mathbf{r}_{t-1}] = \mathbf{k}_t(\boldsymbol{z})^\top \boldsymbol{\alpha}_t$$

$$\mathbf{Cov}[Q(\boldsymbol{z}), Q(\boldsymbol{z}') | R_{t-1} = \mathbf{r}_{t-1}] = k(\boldsymbol{z}, \boldsymbol{z}') - \mathbf{k}_t(\boldsymbol{z})^\top \mathbf{C}_t \mathbf{k}_t(\boldsymbol{z}')$$

Bayes

RL

Policy Improvement

Optimistic Policy Iteration algorithms work by maintaining a policy evaluator $\hat{Q}_t$ and selecting the action at time t semi-greedily w.r.t. to the current state-action value estimates $\hat{Q}_t(\boldsymbol{x}_t, \cdot)$.

Policy evaluator	Parameters	OPI algorithm
Online TD(λ) (Sutton)	$\mathbf{w}_t$	SARSA (Rummery & Niranjan)
Online GPTD (Engel et Al.)	$\boldsymbol{\alpha}_t$, $\mathbf{C}_t$	GPSARSA (Engel et Al.)

GPSARSA ALGORITHM

Initialize $\boldsymbol{\alpha}_0 = \mathbf{0}$, $\mathbf{C}_0 = 0$, $\mathcal{D}_0 = \{\boldsymbol{z}_0\}$, $\mathbf{c}_0 = \mathbf{0}$, $d_0 = 0$, $1/s_0 = 0$

for $t = 1, 2, \ldots$

  observe $\boldsymbol{x}_{t-1}$, $\boldsymbol{a}_{t-1}$, r_{t-1}, $\boldsymbol{x}_t$

  $\boldsymbol{a}_t = \textbf{SemiGreedyAction}(\boldsymbol{x}_t, \mathcal{D}_{t-1}, \boldsymbol{\alpha}_{t-1}, \mathbf{C}_{t-1})$

  $d_t = \frac{\gamma\sigma^2_{t-1}}{s_{t-1}} d_{t-1} + \text{temporal difference}$

  $\mathbf{c}_t = \ldots, \quad s_t = \ldots$

$$\boldsymbol{\alpha}_t = \begin{pmatrix} \boldsymbol{\alpha}_{t-1} \\ 0 \end{pmatrix} + \frac{\mathbf{c}_t}{s_t} d_t$$

$$\mathbf{C}_t = \begin{bmatrix} \mathbf{C}_{t-1} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} + \frac{1}{s_t} \mathbf{c}_t \mathbf{c}_t^\top$$

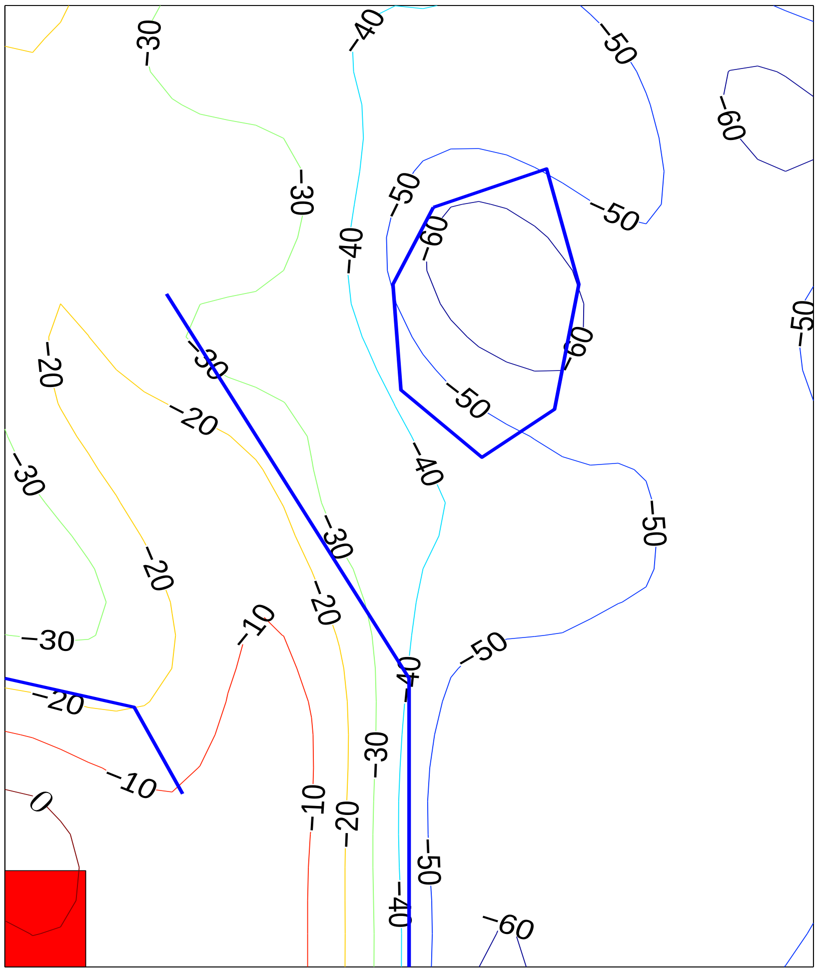
  $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{\boldsymbol{z}_t\}$

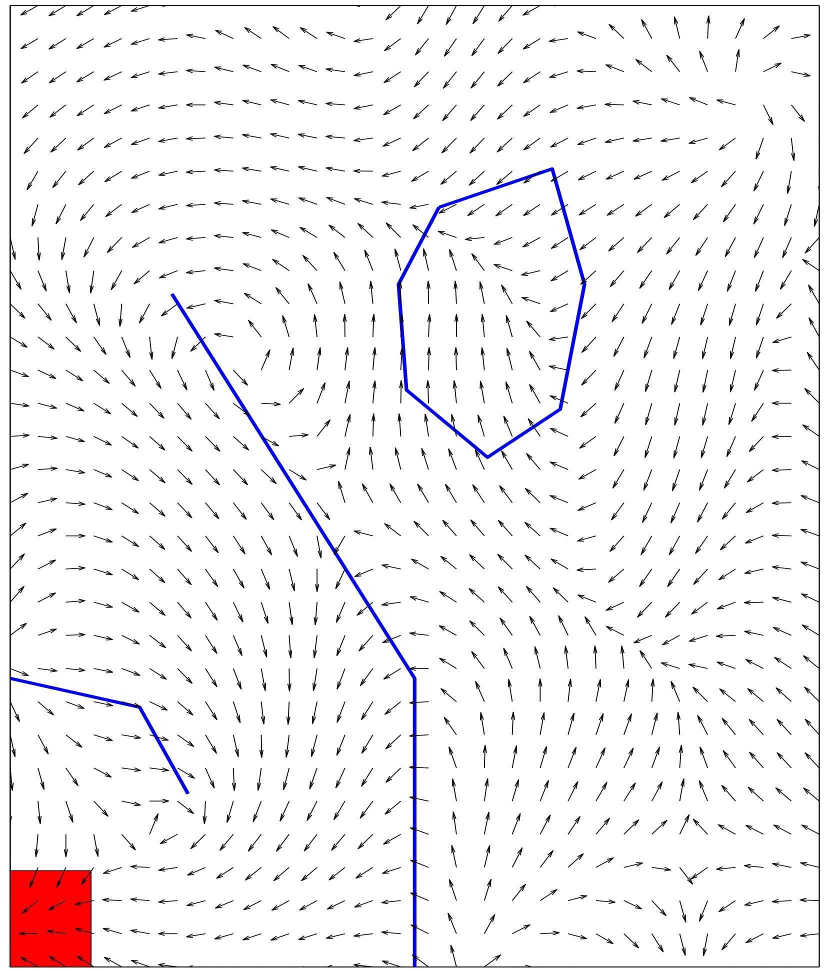
end for

return $\boldsymbol{\alpha}_t$, $\mathbf{C}_t$, $\mathcal{D}_t$

Bayes RL

A 2D Navigation Task





Challenges

- How to use value uncertainty?
- What's a disciplined way to select actions?
- What's the best noise covariance?
- Bias, variance, learning curves
- POMDPs
- More complicated tasks

Questions?