

Reinforcement Learning with Gaussian Processes

Shie Mannor
McGill University



McGill

Joint work with Yaakov Engel (U. Alberta) and Ron Meir (Technion)

THE BAYESIAN RELIGION



- Z – hidden process, Y – observable
- We want to infer Z from measurements of Y
- Statistical dependence between Z and Y known: $P(Y|Z)$
- Place prior over Z , reflecting our uncertainty: $P(Z)$
- Observe $Y = y$
- Compute posterior: $P(Z|Y = y) = \frac{P(y|Z)P(Z)}{\int dZ' P(y|Z')P(Z')}$

WHY USE GPs IN RL?

- A Bayesian approach to value estimation
- Non-parametric – priors are placed and inference is performed directly in **function** space (kernels).
- Domain knowledge intuitively coded into priors
- Provides full posterior, not just point estimates
- Efficient, on-line implementation, suitable for large problems on complicated spaces

GAUSSIAN PROCESSES 101

Definition: “An **indexed** set of jointly Gaussian random variables”

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

Example: $F(\mathbf{x})$, index is $\mathbf{x} \in$ possible inventories

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

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

m is a function $\mathcal{X} \rightarrow \mathbb{R}$, k is a function $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.

Conditions on k :

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

GP REGRESSION

Model equation:

$$Y(\mathbf{x}) = F(\mathbf{x}) + N(\mathbf{x})$$

Prior:

$$F \sim \mathcal{N}\{0, k(\cdot, \cdot)\}$$

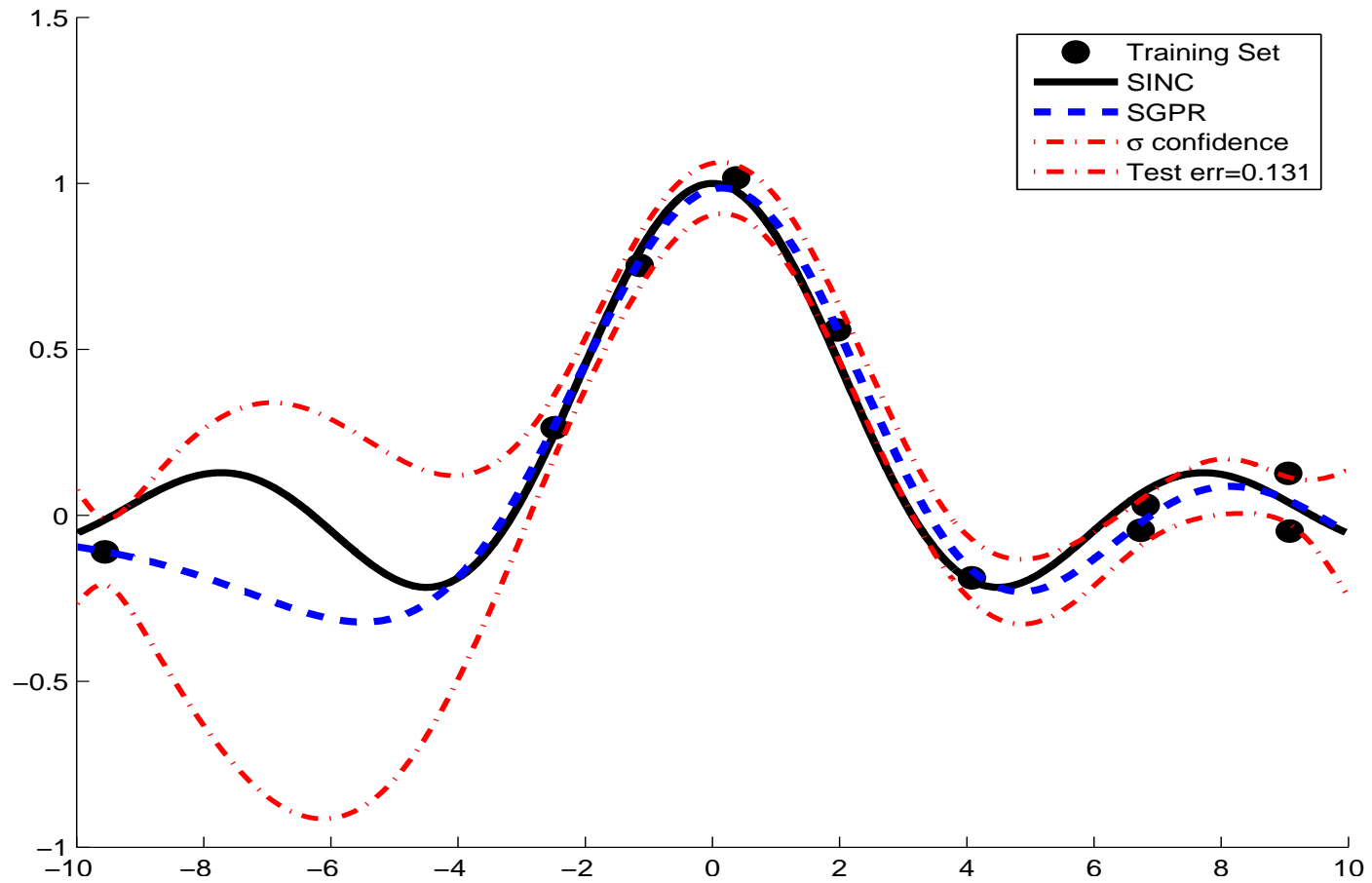
Noise:

$$N \sim \mathcal{N}\{0, \sigma^2 \delta(\cdot - \cdot)\}$$

Goal:

Find the **posterior** distribution of F ,
given a sample for Y (via Bayes' rule)

EXAMPLE



MARKOV DECISION PROCESSES

\mathcal{X} : state space

\mathcal{U} : action space

$p: \mathcal{X} \times \mathcal{X} \times \mathcal{U} \rightarrow [0, 1], \quad \mathbf{x}_{t+1} \sim p(\cdot | \mathbf{x}_t, \mathbf{u}_t)$

$q: \mathbb{R} \times \mathcal{X} \times \mathcal{U} \rightarrow [0, 1], \quad R(\mathbf{x}_t, \mathbf{u}_t) \sim q(\cdot | \mathbf{x}_t, \mathbf{u}_t)$

A Stationary policy:

$\mu: \mathcal{U} \times \mathcal{X} \rightarrow [0, 1], \quad \mathbf{u}_t \sim \mu(\cdot | \mathbf{x}_t)$

Discounted Return: $D^\mu(\mathbf{x}) = \sum_{i=0}^{\infty} \gamma^i R(\mathbf{x}_i, \mathbf{u}_i) | (\mathbf{x}_0 = \mathbf{x})$

Value function: $v^\mu(\mathbf{x}) = \mathbb{E}_\mu[D^\mu(\mathbf{x})]$

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

THE DISCOUNTED RETURN

A Random Process:

$$D^\mu(\mathbf{x}) = \sum_{i=0}^{\infty} \gamma^i R(\mathbf{x}_i, \mathbf{u}_i) \Big| (\mathbf{x}_0 = \mathbf{x})$$

Therefore: $D^\mu(\mathbf{x}) = R(\mathbf{x}, \mathbf{u}) + \gamma D^\mu(\mathbf{x}')$

Where: $\mathbf{u} \sim \mu(\cdot|\mathbf{x})$ and $\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})$

Also (trivially):

$$D^\mu(\mathbf{x}) = v^\mu(\mathbf{x}) + (D^\mu(\mathbf{x}) - v^\mu(\mathbf{x}))$$

OUR APPROACH

Classical approach: look for the **function** $v^\mu(\mathbf{x})$. I.e.,

$$D^\mu(\mathbf{x}) = v^\mu(\mathbf{x}) + \Delta V^\mu(\mathbf{x})$$

GP approach: the value is also a **random variable**. I.e.,

$$D^\mu(\mathbf{x}) = V^\mu(\mathbf{x}) + \Delta V^\mu(\mathbf{x})$$

Value function $v^\mu(x) = \mathbb{E}_{\text{models}} [V_{\text{model}}^\mu(\mathbf{x})]$

By assuming a Gaussian structure $v^\mu(x)$ is easy to compute.

A GENERATIVE MODEL FOR VALUES

The generative model:

$$\begin{aligned}R(\mathbf{x}_i, \mathbf{u}_i) &= V(\mathbf{x}_i) - \gamma V(\mathbf{x}_{i+1}) + N(\mathbf{x}_i, \mathbf{x}_{i+1}) \\ &= H(\mathbf{x}_i, \mathbf{x}_{i+1})V + N(\mathbf{x}_i, \mathbf{x}_{i+1}) \\ V &\sim \mathcal{N}\{0, k(\cdot, \cdot)\}\end{aligned}$$

H is a **linear** integral operator defined by:

$$H(\mathbf{x}, \mathbf{x}')V = \int d\mathbf{s} (\delta(\mathbf{s} - \mathbf{x}) - \gamma\delta(\mathbf{s} - \mathbf{x}')) V(\mathbf{s})$$

Goal:

Find the posterior distribution of $V(\cdot)$, given a sequence of observed states and rewards

DYNAMICS

Model Equations:

$$\sum_{j=i}^t \gamma^{j-i} R(\mathbf{x}_j) = D(\mathbf{x}_i) = V(\mathbf{x}_i) + \Delta V(\mathbf{x}_i)$$

For a stationary MDP:

$$D(\mathbf{x}_i) = R(\mathbf{x}_i, \mathbf{u}_i) + \gamma D(\mathbf{x}_{i+1}) \quad (\mathbf{u}_i \sim \mu(\cdot | \mathbf{x}_i), \mathbf{x}_{i+1} \sim p(\cdot | \mathbf{x}_i, \mathbf{u}_i))$$

Substitute and rearrange:

$$\begin{aligned} R(\mathbf{x}_i, \mathbf{u}_i) &= V(\mathbf{x}_i) - \gamma V(\mathbf{x}_{i+1}) + N(\mathbf{x}_i, \mathbf{x}_{i+1}) \\ N(\mathbf{x}_i, \mathbf{x}_{i+1}) &\stackrel{\text{def}}{=} \Delta V(\mathbf{x}_i) - \gamma \Delta V(\mathbf{x}_{i+1}) \end{aligned}$$

We end up with:

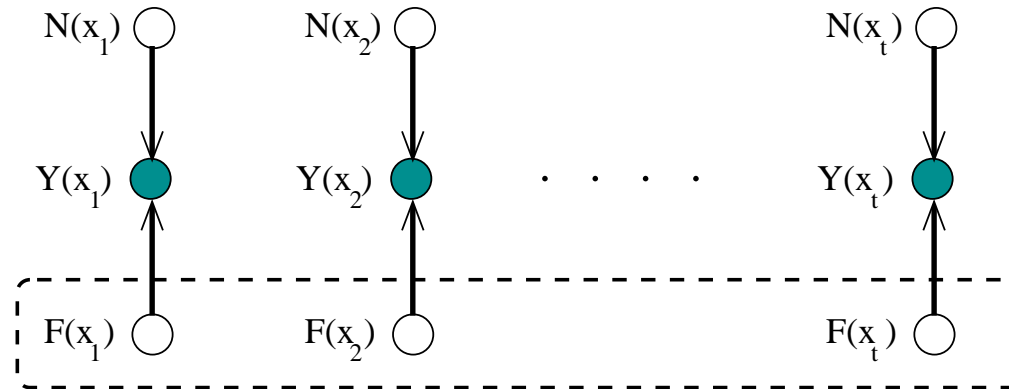
$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t, \text{ with } N_t \sim \mathcal{N} \{0, \sigma^2 \mathbf{H}_{t+1} \mathbf{H}_{t+1}^\top\}$$

THE GPTD STATISTICAL MODEL

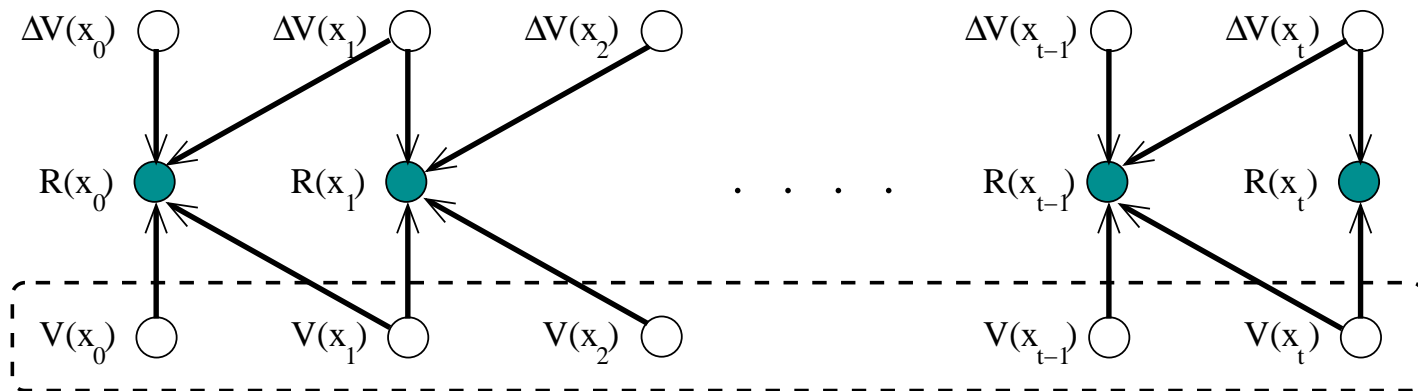
	GP Regression	GPTD
Latent GP	F	V
Observable	Y	R
Gen. model	$Y_t = F_t + N_t$	$R_t = \mathbf{H}_{t+1}V_{t+1} + N_t$
		$\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ 0 & 1 & -\gamma & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 1 & -\gamma \end{bmatrix}$
Noise Stat.	Gaussian, white	Gaussian, correlated
Noise Cov	$\sigma^2\mathbf{I}$	$\Sigma_t = \sigma^2\mathbf{H}_{t+1}\mathbf{H}_{t+1}^\top$

THE GPTD STATISTICAL MODEL (CTD.)

GP Regression graph



GPTD graph



THE POSTERIOR

General noise covariance:

$$\text{Cov}[N_t] = \Sigma_t$$

Joint distribution:

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

Invoke the Gauss-Markov Theorem:

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

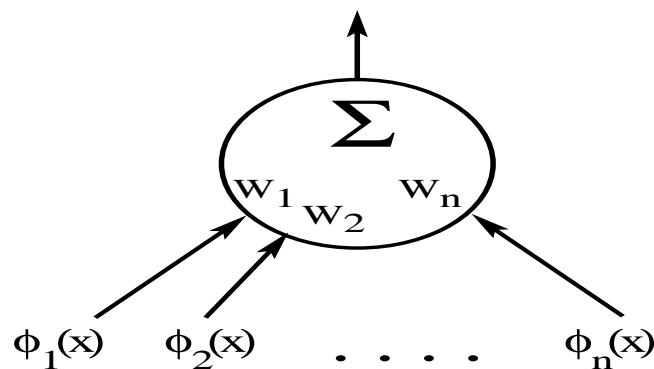
$$\text{Cov}[V(\mathbf{x}), V(\mathbf{x}') | R_{t-1}] = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_t(\mathbf{x})^\top \mathbf{C}_t \mathbf{k}_t(\mathbf{x}')$$

$$\mathbf{k}_t(\mathbf{x}) = (k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_t, \mathbf{x}))^\top$$

A PARAMETRIC GAUSSIAN PROCESS MODEL

A linear combination of features:

$$V(\mathbf{x}) = \phi(\mathbf{x})^\top W$$



Prior on W : Gaussian, with $\mathbb{E}[W] = \mathbf{0}$, $\text{Cov}[W, W] = \mathbf{I}$

Prior on V : Gaussian, with

$$\mathbb{E}[V(\mathbf{x})] = \mathbf{0}, \quad \text{Cov}[V(\mathbf{x}), V(\mathbf{x}')] = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$$

NONPARAMETRIC GPTD

	Parametric	Nonparametric
Parametrization	$V(\mathbf{x}) = \phi(\mathbf{x})^\top W$	None, V is V
Prior	$W \sim \mathcal{N}\{\mathbf{0}, \mathbf{I}\}$	$V \sim \mathcal{N}\{0, k(\cdot, \cdot)\}$
$\mathbb{E}[V(\mathbf{x})]$	0	0
$\mathbb{E}[V(\mathbf{x})V(\mathbf{x}')]]$	$\phi(\mathbf{x})^\top \phi(\mathbf{x}')$	$k(\mathbf{x}, \mathbf{x}')$
We seek	$W R_{t-1}$	$V(\mathbf{x}) R_{t-1}$

If we can find a set of basis functions satisfying $\phi(\mathbf{x})^\top \phi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$ then the two priors become equivalent.

In fact, such a set **always** exists [Mercer].

However, it may be infinite!

RELATION TO MONTE-CARLO ESTIMATION

In the stochastic model: $\Sigma_t = \sigma^2 \mathbf{H}_{t+1} \mathbf{H}_{t+1}^\top$

Also, let: $(Y_t)_i = \sum_{j=i}^t \gamma^{j-i} R(\mathbf{x}_j, \mathbf{u}_j)$

Then:

$$\begin{aligned}\mathbb{E}[W|R_t] &= \left(\Phi_t \Phi_t^\top + \sigma^2 \mathbf{I} \right)^{-1} \Phi_t Y_t \\ \text{Cov}[W|R_t] &= \sigma^2 \left(\Phi_t \Phi_t^\top + \sigma^2 \mathbf{I} \right)^{-1}\end{aligned}$$

That's the solution to GP regression on Monte-Carlo samples of the discounted return.

ON-LINE OPERATION

In the parametric case:

Kalman Filter updates, without the “state” dynamics (i.e.,

$$W_{t+1} = W_t)$$

Cost per time-step: $O(n^2)$

In the nonparametric case:

Nonparametric Kalman-like updates

Cost per time-step:

With sparsification, $O(m^2)$, where m is the **dictionary** size.

WRAP-UP

- It's good to be Bayesian
- Completely arbitrary state space
- Policy improvement a-la-SARSA
- Design a kernel, not basis functions
 - Rich representations
 - Decomposable/hierarchical kernels
- Extrinsic variance is useful (exploration, termination conditions, etc.)
- Convergence rates
- Learning is **not** based on decreasing learning rates