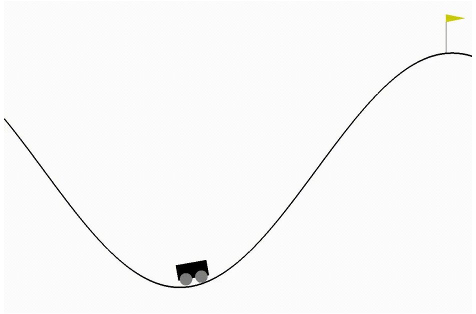
***Gaussian Process** is*
All You Need



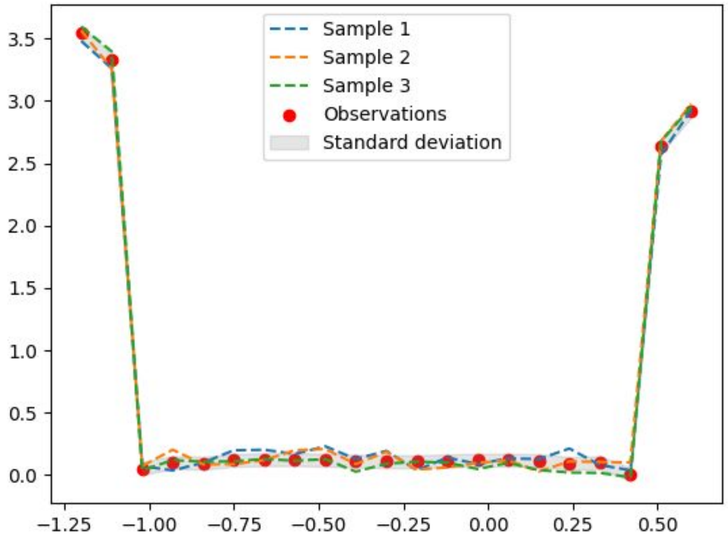
- GP introduction
- Simulation Results
- Algorithm comparisons
- Conclusion

Presented By

Steve Wen
Chuqiao(Chloe) Si

Introduction to Gaussian Process

- **Gaussian Process(GP)** is a probabilistic perspective in predicting the value of functions at new data points, given observations at other points.



A slice of the GP value function where v= 0

Any condition and marginalization of a Gaussian distribution is also Gaussian:

$$
\begin{aligned}
& p(y^*|\mathbf{x}^*, X, \mathbf{y}, \mathbf{w}) \\
& = \int p(y^*, \mathbf{w}|\mathbf{x}^*, X, \mathbf{y}) d\mathbf{w} \\
& = \int p(y^*|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} \\
& \sim \mathcal{N}(k(\mathbf{x}^*, X)\mathbf{K}^{-1}\mathbf{y}, k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, X)\mathbf{K}^{-1}k(X, \mathbf{x}^*))
\end{aligned}
$$

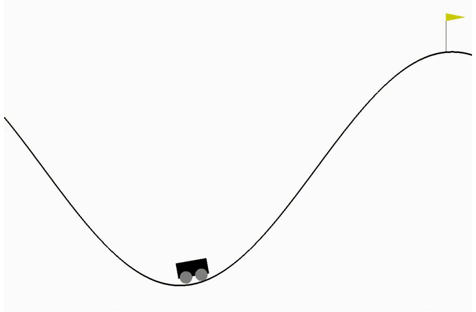
Gaussian Process in Reinforcement Learning Algorithm

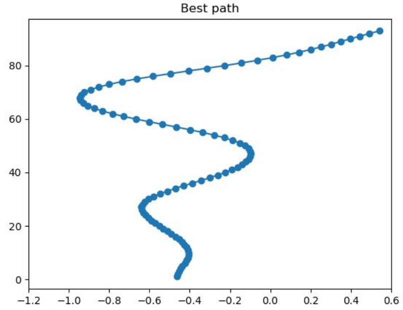
Algorithm 1 GPRL value function iteration with greedy action in a deterministic environment

Initialize an $N \times N$ grid $G = \{(S_1, V(S_1)), \dots (S_{N\times N}, V(S_{N\times N}))\}$ to partition the state space with environment given reward as its initial value V, given hyperparameters γ, T, N.
Fit GP using G as the supporting points.
for $t \in \{1, \dots T\}$ **do**
    for $s \in \{S_1, \dots S_{N\times N}\}$ **do**
        Find the greedy action a and calculate s', r based on deterministic dynamics
        Use GP to predict $V(s')$
        Update $V(s) = r + \gamma V(s')$
    end for
    Refit GP using the updated grid values
end for

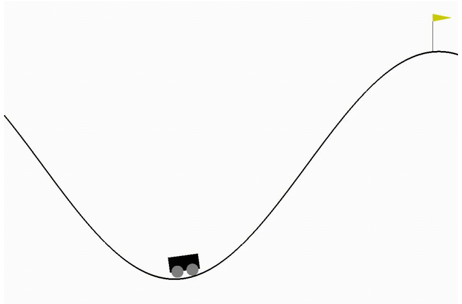
Results—best path and simulation (50 iterations)

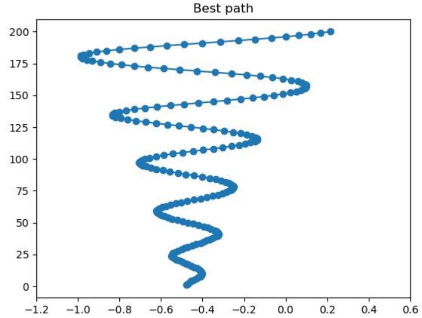
Kernel: Matern



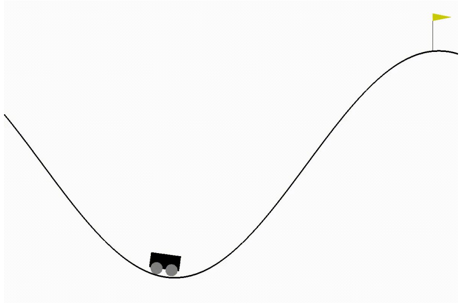


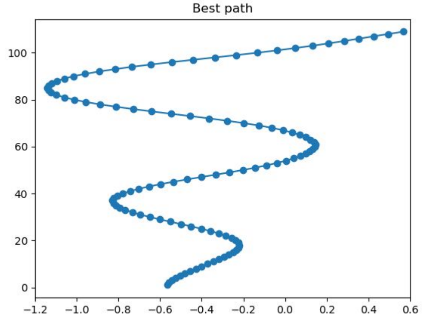
Kernel: RBF



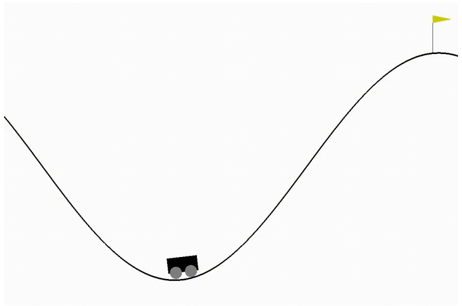


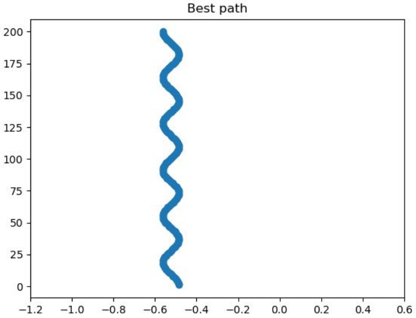
Kernel: RationalQuadratic





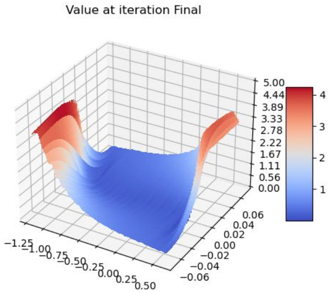
Kernel: ExpSineSquared



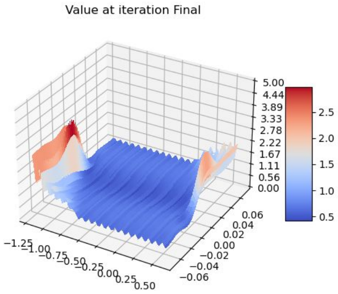


Results—value function surface after 50 iterations

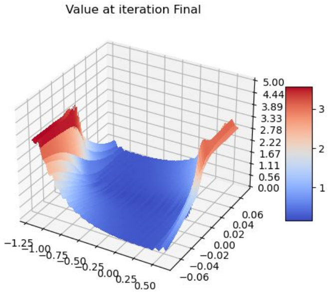
Kernel: Matern



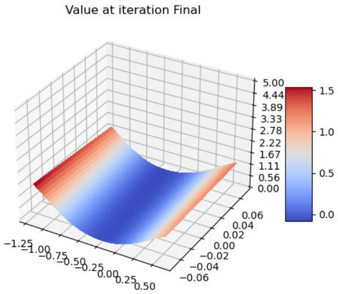
Kernel: RBF

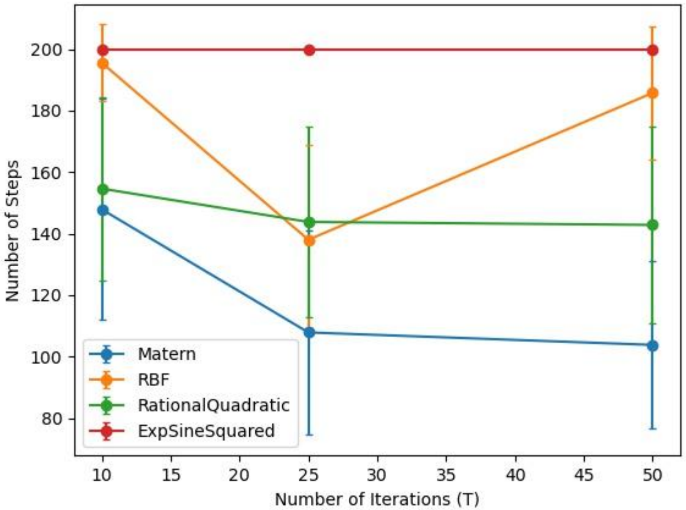


Kernel: RationalQuadratic



Kernel: ExpSineSquared





Kernel, Algorithm Comparison & Conclusion

Table 1: GP against other algorithms on mountain car

Algorithms	Avg. Steps	Avg. Uncertainty
GP(Matern)	**103.8 ± 27.3**	**0.05**
Q-Learning	178.2 ± 12.6	0.28
DQN	127.9 ± 23.5	4.57
DP	200 ± 0	—

- Gaussian Process outperforms other trending RL algorithms on mountain car task by a large margin and is more data efficient
- Gaussian Process has low uncertainty in value function evaluations
- Kernel choices are important to the performance of Gaussian Process
- Number of value function iterations does matter