On Kernelized Multi-armed Bandits

Sayak Ray Chowdhury Aditya Gopalan

Department of Electrical Communication Engineering
Indian Institute of Science

ICML
August 7, 2017

Overview

Problem Formulation

Algorithms

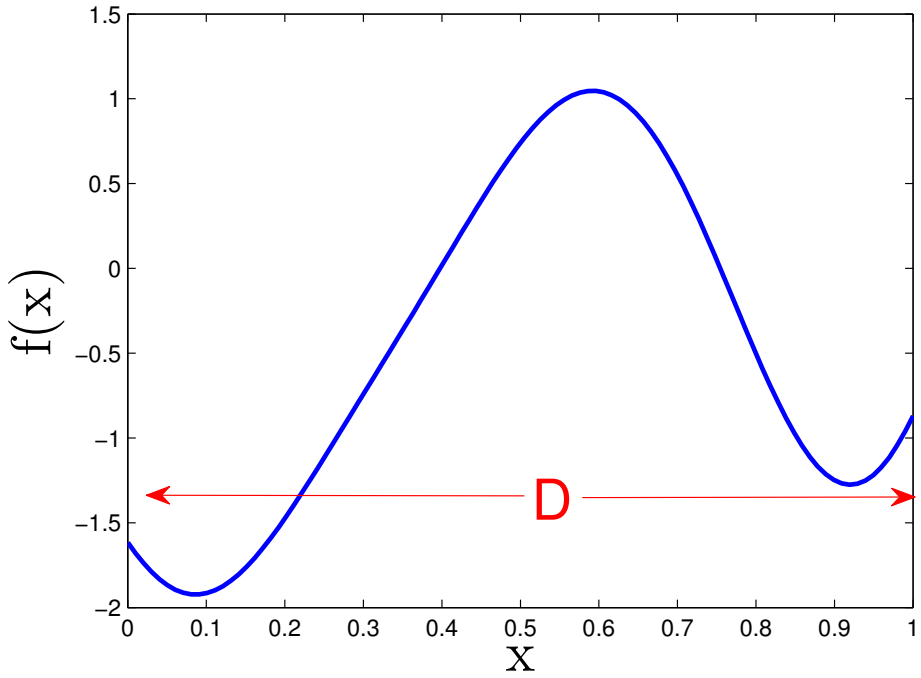
Regret Bounds

Numerical Results

Conclusion

Problem Statement

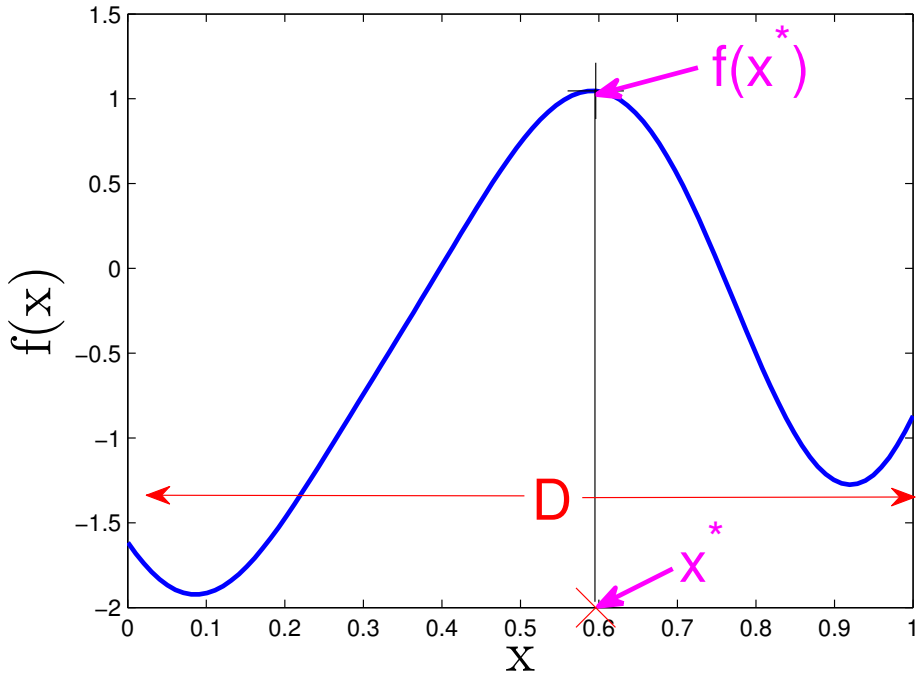
Sequentially Maximize $f : D \to \mathbb{R}$



- f unknown, $D \subset \mathbb{R}^d$

Problem Statement

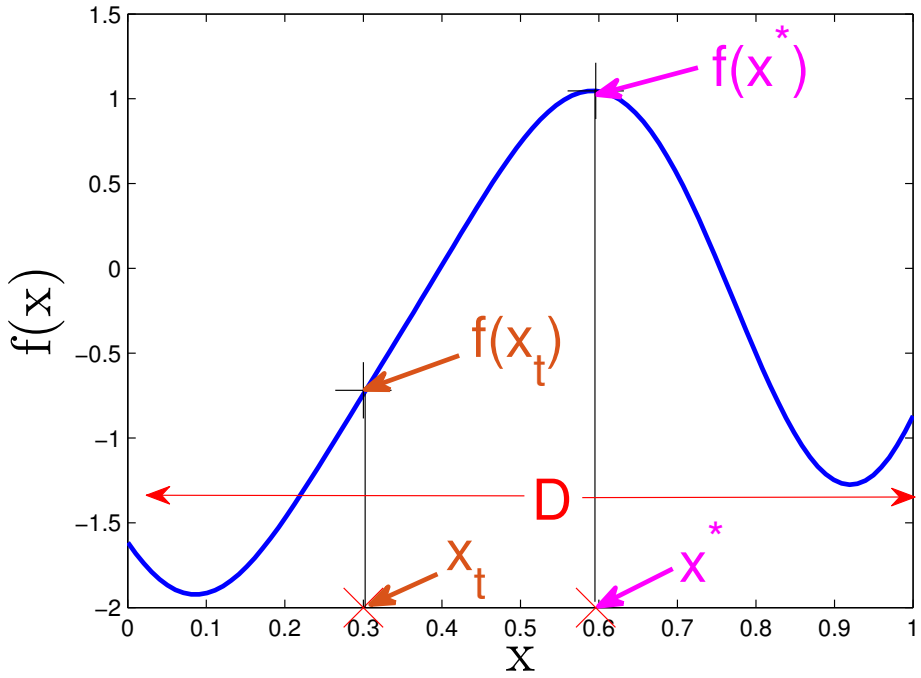
Sequentially Maximize $f : D \to \mathbb{R}$



- f unknown, $D \subset \mathbb{R}^d$
- $x^\star = \underset{x \in D}{\operatorname{argmax}}\, f(x)$

Problem Statement

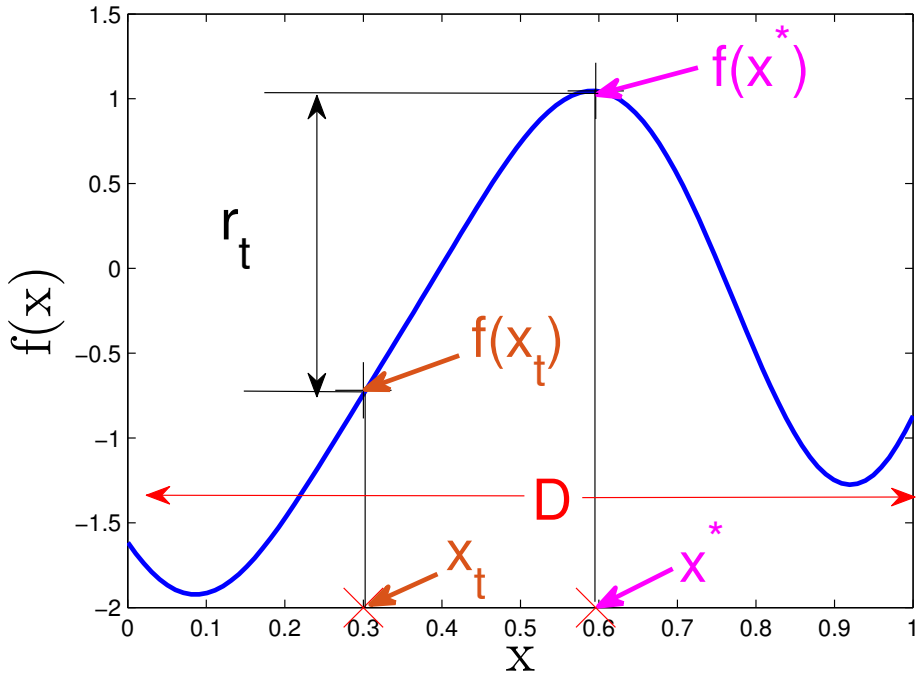
Sequentially Maximize $f : D \to \mathbb{R}$



- f unknown, $D \subset \mathbb{R}^d$
- $x^\star = \underset{x \in D}{\operatorname{argmax}}\, f(x)$
- At each round t:
  - Learner chooses $x_t \in D$ based on past
  - Observes noisy reward $y_t = f(x_t) + \varepsilon_t$

Problem Statement

Sequentially Maximize $f : D \to \mathbb{R}$



- f unknown, $D \subset \mathbb{R}^d$
- $x^\star = \underset{x \in D}{\operatorname{argmax}}\, f(x)$
- At each round t:
  - Learner chooses $x_t \in D$ based on past
  - Observes noisy reward $y_t = f(x_t) + \varepsilon_t$

Performance Metric

- **Regret** $r_t = f(x^\star) - f(x_t)$
- Goal: Minimize cumulative regret $\sum_{t=1}^{T} r_t$

Assumptions

- Noise ε_t is R-sub-Gaussian

Assumptions

- Noise ε_t is R-sub-Gaussian

- f lies in **RKHS** of functions: $D \to \mathbb{R}$
- Positive semi-definite kernel function $k : D \times D \to \mathbb{R}$ (known)
- Reproducing property: $f(x) = \langle f, k(x, \cdot) \rangle_k$
- Induces smoothness: $|f(x) - f(y)| \leq \|f\|_k \, \|k(x, \cdot) - k(y, \cdot)\|_k$

Assumptions

- Noise ε_t is R-sub-Gaussian

- f lies in **RKHS** of functions: $D \to \mathbb{R}$
- Positive semi-definite kernel function $k : D \times D \to \mathbb{R}$ (known)
- Reproducing property: $f(x) = \langle f, k(x, \cdot) \rangle_k$
- Induces smoothness: $|f(x) - f(y)| \leq \|f\|_k \, \|k(x, \cdot) - k(y, \cdot)\|_k$

- D is compact, $\|f\|_k \leq B$ known

Assumptions

- Noise ε_t is R-sub-Gaussian
- f lies in **RKHS** of functions: $D \to \mathbb{R}$
- Positive semi-definite kernel function $k : D \times D \to \mathbb{R}$ (known)
- Reproducing property: $f(x) = \langle f, k(x, \cdot) \rangle_k$
- Induces smoothness: $|f(x) - f(y)| \leq \|f\|_k \, \|k(x, \cdot) - k(y, \cdot)\|_k$
- D is compact, $\|f\|_k \leq B$ known
- Bounded variance: $k(x, x) \leq 1$, for all $x \in D$

Example Kernels

- Squared Exponential kernel: $k(x, y) = \exp\left(\frac{-\|x-y\|_2^2}{2l^2}\right)$
- Matérn kernel: $k(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)^{\nu} B_\nu\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)$
- **Stationary** kernels: $k(x, y) \equiv k(x - y)$

Example Kernels

- Squared Exponential kernel: $k(x, y) = \exp\left(\frac{-\|x-y\|_2^2}{2l^2}\right)$
- Matérn kernel: $k(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)^{\nu} B_\nu\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)$
- **Stationary** kernels: $k(x, y) \equiv k(x - y)$

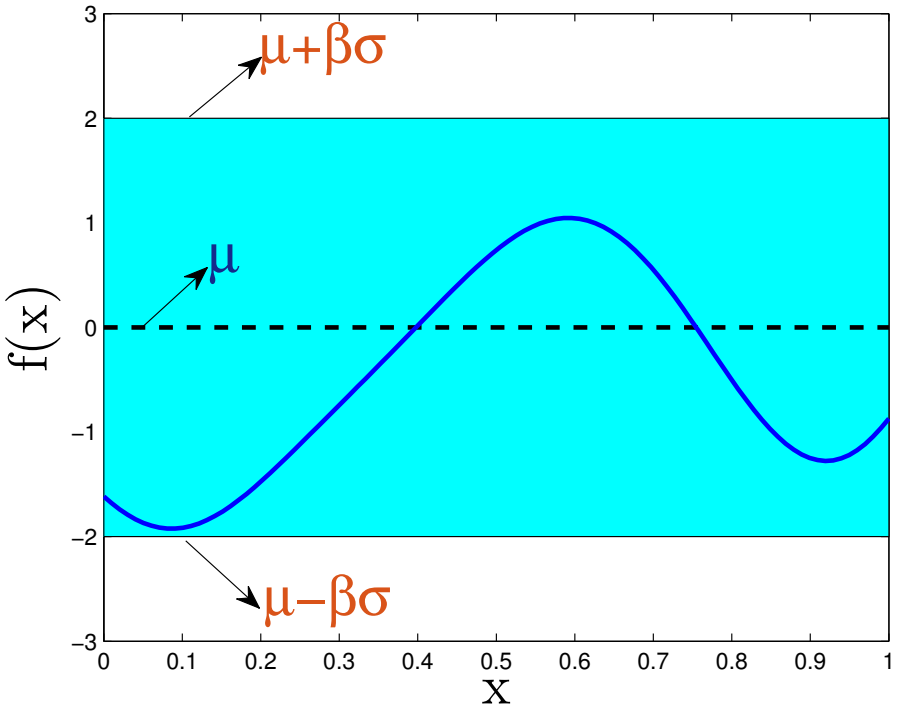
- Linear Kernel:
  - $k(x, y) = x^T y$
  - $f(x) = \theta^T x$, $\theta \in \mathbb{R}^d$ unknown parameter

Example Kernels

- Squared Exponential kernel: $k(x, y) = \exp\left(\frac{-\|x-y\|_2^2}{2l^2}\right)$
- Matérn kernel: $k(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)^{\nu} B_\nu\left(\frac{\|x-y\|_2\sqrt{2\nu}}{l}\right)$
- **Stationary** kernels: $k(x, y) \equiv k(x - y)$

- Linear Kernel:
  - $k(x, y) = x^T y$
  - $f(x) = \theta^T x$, $\theta \in \mathbb{R}^d$ unknown parameter
  - Reduces to parametric **linear bandit** problem (Dani et al., COLT 2008, Abbasi-Yadkori et al., NIPS 2011, ...)

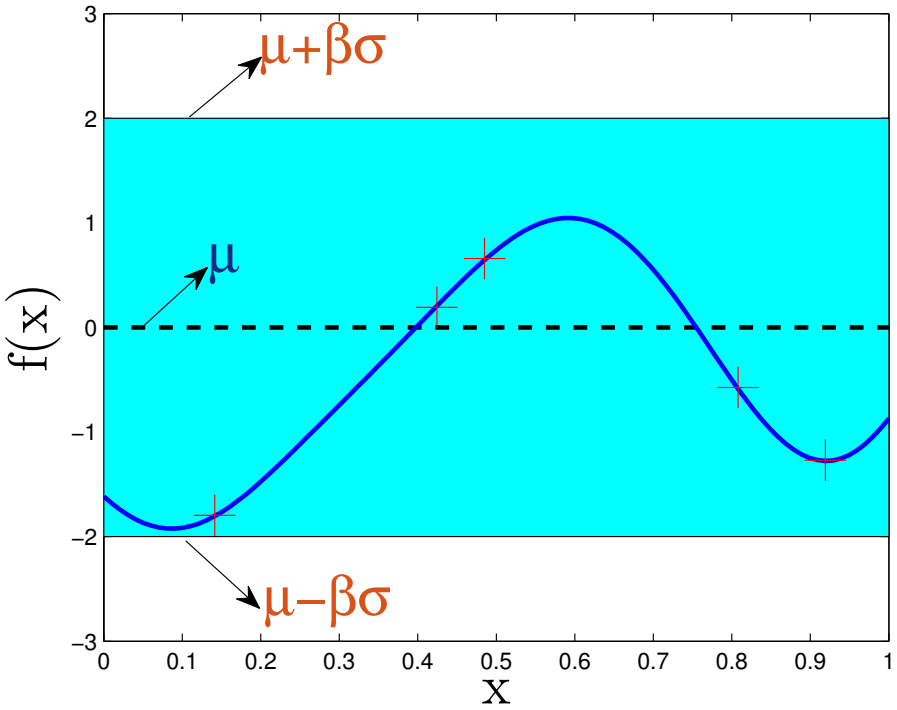
Algorithm Design Philosophy: Gaussian Processes



Assume:

- Gaussian Process Prior of f: $GP(0, v^2 k(x, y))$
- Noise $\varepsilon_t \sim \mathcal{N}(0, \lambda v^2)$

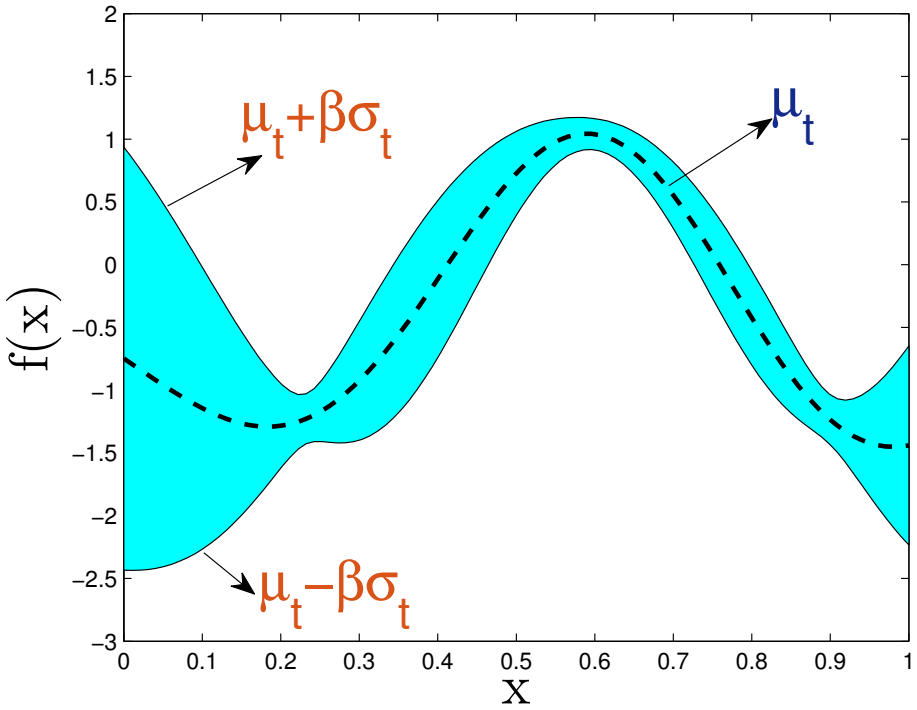
Algorithm Design Philosophy: Gaussian Processes



Assume:

- Gaussian Process Prior of f: $GP(0, v^2 k(x, y))$
- Noise $\varepsilon_t \sim \mathcal{N}(0, \lambda v^2)$
- After t rounds, reward vector $y_{1:t} \sim \mathcal{N}(0, v^2(K_t + \lambda I))$

Algorithm Design Philosophy: Gaussian Processes



Assume:

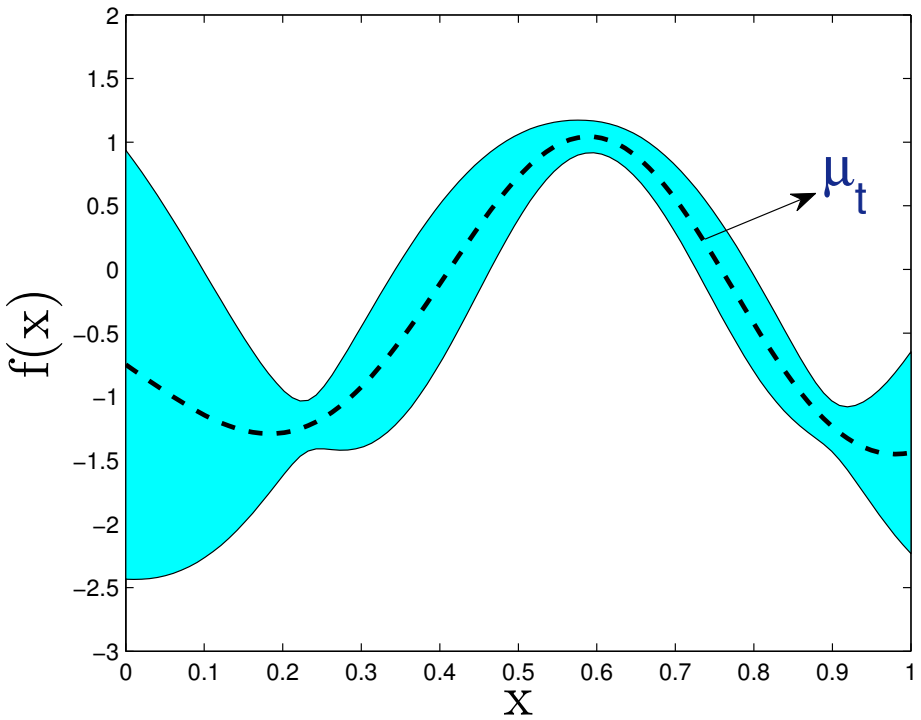
- Gaussian Process Prior of f: $GP(0, v^2 k(x,y))$
- Noise $\varepsilon_t \sim \mathcal{N}(0, \lambda v^2)$
- After t rounds, reward vector $y_{1:t} \sim \mathcal{N}(0, v^2(K_t + \lambda I))$

Posterior of f after t rounds: $GP(\mu_t(x), v^2 k_t(x,y))$

$$\begin{aligned} \mu_t(x) &= k_t(x)^T (K_t + \lambda I)^{-1} y_{1:t} \\ k_t(x,y) &= k(x,y) - k_t(x)^T (K_t + \lambda I)^{-1} k_t(y) \end{aligned}$$

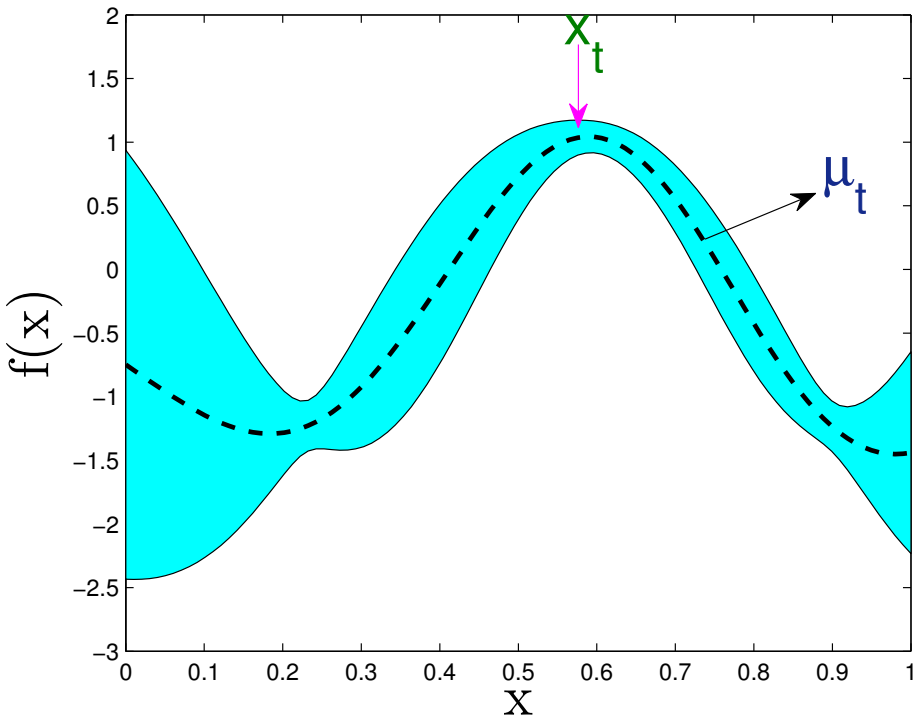
Algorithm 1: Improved GP-UCB (IGP-UCB)

Key Idea: Play the arm with highest UCB



Algorithm 1: Improved GP-UCB (IGP-UCB)

Key Idea: Play the arm with highest UCB

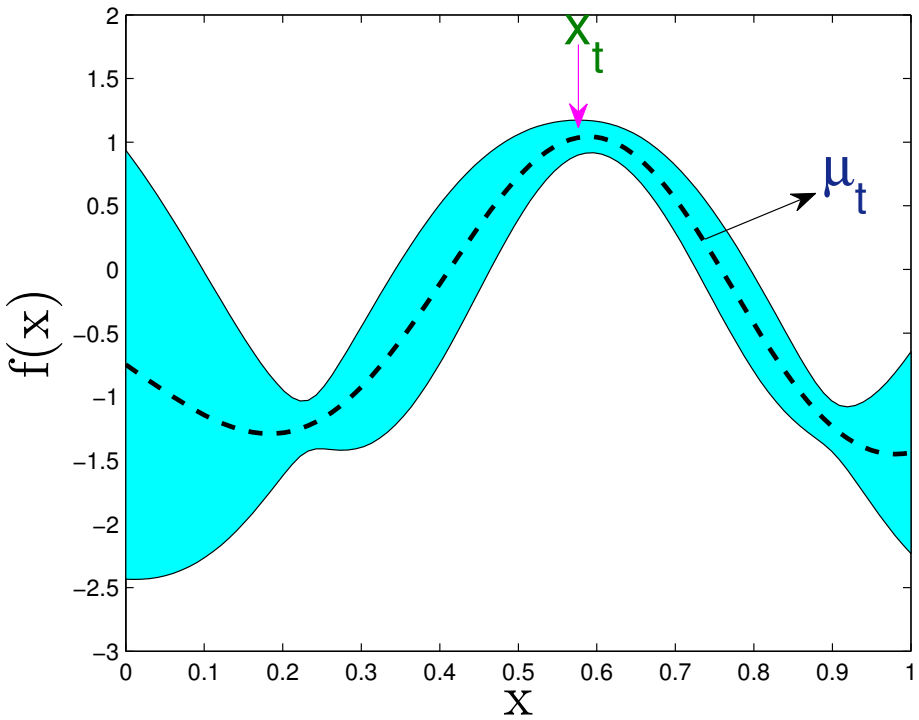


At each round t, play:

$$x_t = \operatorname*{argmax}_{x \in D} \mu_t(x) + \beta_t \sigma_t(x)$$

Algorithm 1: Improved GP-UCB (IGP-UCB)

Key Idea: Play the arm with highest UCB



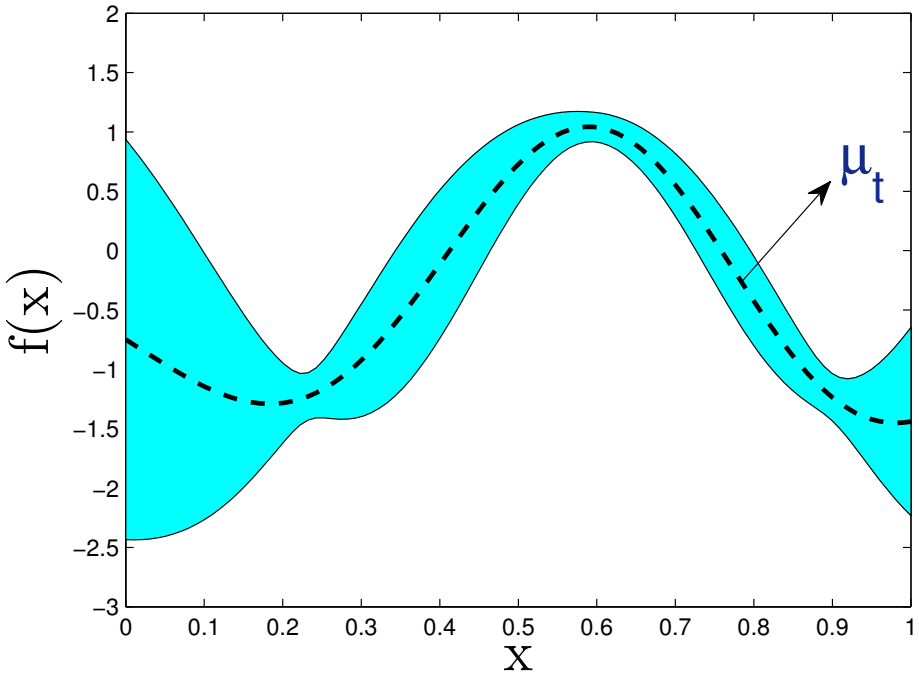
At each round t, play:

$$x_t = \operatorname*{argmax}_{x \in D} \mu_t(x) + \beta_t \sigma_t(x)$$

- β_t trades off b/w **exploration** and **exploitation**
- Reduced width (β_t) of confidence interval compared to GP-UCB (Srinivas et al., ICML 2010)

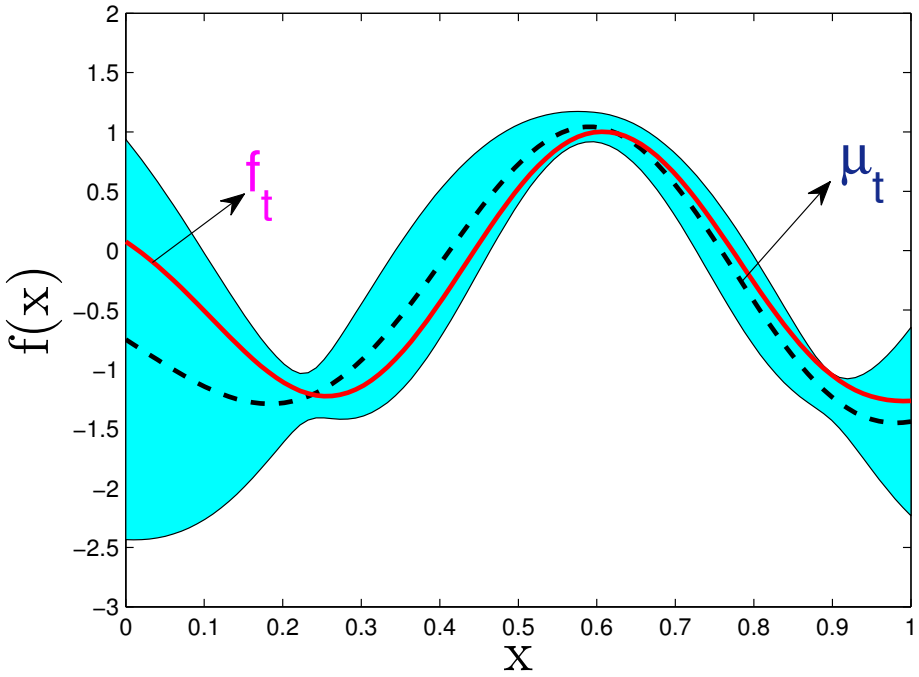
Algorithm 2: Gaussian Process Thompson Sampling (GP-TS)

Key Idea: Sample a random function and play its maximizer



Algorithm 2: Gaussian Process Thompson Sampling (GP-TS)

Key Idea: Sample a random function and play its maximizer

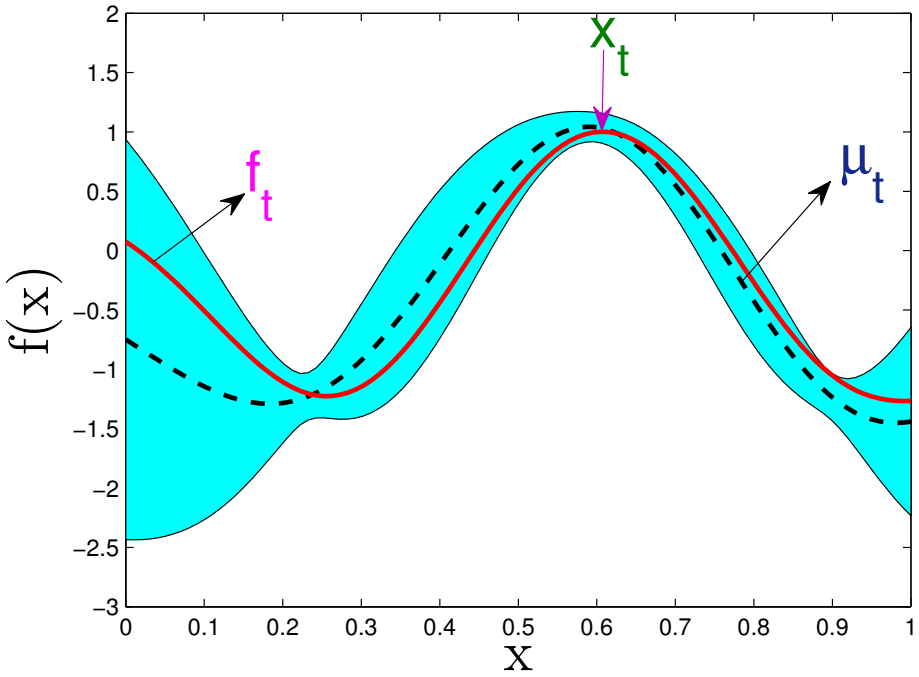


At each round t:

- Sample f_t from posterior of f

Algorithm 2: Gaussian Process Thompson Sampling (GP-TS)

Key Idea: Sample a random function and play its maximizer



At each round t:

- Sample f_t from posterior of f
- Play $x_t = \operatorname*{argmax}_{x \in D_t} f_t(x)$

$D_t \subset D$: suitably chosen Discretization sets

Regret Bound for IGP-UCB

Result 1

Regret of IGP-UCB is $O\left(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T)\right)$ whp with the choice of confidence width $\beta_t \approx B + \sqrt{\gamma_t}$ for all t

Regret Bound for IGP-UCB

Result 1

Regret of IGP-UCB is $O\left(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T)\right)$ whp with the choice of confidence width $\beta_t \approx B + \sqrt{\gamma_t}$ for all t

- γ_T is **Maximum Information Gain** after T rounds:

$$\gamma_T = \max_{A \subset D : |A| = T} I(y_A; f_A)$$

  - Mutual Information b/w function values and rewards at set A
  - Reduction in uncertainty about f after observing rewards
  - SE kernel: $\gamma_T = O((\ln T)^{d+1}) \rightarrow$ **sublinear** regret

Regret Bound for IGP-UCB

Result 1

Regret of IGP-UCB is $O\left(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T)\right)$ whp with the choice of confidence width $\beta_t \approx B + \sqrt{\gamma_t}$ for all t

- γ_T is **Maximum Information Gain** after T rounds:

$$\gamma_T = \max_{A \subset D : |A| = T} I(y_A; f_A)$$

  - Mutual Information b/w function values and rewards at set A
  - Reduction in uncertainty about f after observing rewards
  - SE kernel: $\gamma_T = O((\ln T)^{d+1}) \rightarrow$ **sublinear** regret

- Regret of GP-UCB is $O\left(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T \ln^{3/2} T)\right)$ whp and so we improve by $O(\ln^{3/2} T)$!

Regret Bound for GP-TS

Result 2

- Regret of GP-TS is $O\left(\sqrt{Td\ln(BdT)}(B\sqrt{\gamma_T}+\gamma_T)\right)$ whp
- First **frequentist** regret guarantee of TS in the **non-parametric** setting of infinite action spaces

Regret Bound for GP-TS

Result 2

- Regret of GP-TS is $O\left(\sqrt{Td\ln(BdT)}(B\sqrt{\gamma_T}+\gamma_T)\right)$ whp
- First **frequentist** regret guarantee of TS in the **non-parametric** setting of infinite action spaces

$\sqrt{d\ln(BdT)} \leftarrow$ Consequence of Discretization

Regret Bound for GP-TS

Result 2

- Regret of GP-TS is $O\left(\sqrt{Td\ln(BdT)}(B\sqrt{\gamma_T}+\gamma_T)\right)$ whp
- First **frequentist** regret guarantee of TS in the **non-parametric** setting of infinite action spaces

$\sqrt{d\ln(BdT)} \leftarrow$ Consequence of Discretization

Open Question: Can the logarithmic dependency be removed?

Recovering Regret Bounds for Linear Bandits

Linear Kernel

- $k(x, y) = x^T y$
- $f(x) = \theta^T x$, $\theta \in \mathbb{R}^d$ unknown parameter
- **Maximum Information Gain**: $\gamma_T = O(d \ln T)$
- Regret of IGP-UCB is $\tilde{O}(d\sqrt{T})$ and GP-TS is $\tilde{O}(d^{3/2}\sqrt{T})$

Recovering Regret Bounds for Linear Bandits

Linear Kernel

- $k(x, y) = x^T y$
- $f(x) = \theta^T x$, $\theta \in \mathbb{R}^d$ unknown parameter
- **Maximum Information Gain**: $\gamma_T = O(d \ln T)$
- Regret of IGP-UCB is $\tilde{O}(d\sqrt{T})$ and GP-TS is $\tilde{O}(d^{3/2}\sqrt{T})$

- Exactly recovers regrets of OFUL (Abbasi-Yadkori et al., NIPS 2011) and Linear TS (Agrawal and Goyal, ICML 2013)

Recovering Regret Bounds for Linear Bandits

Linear Kernel

- $k(x, y) = x^T y$
- $f(x) = \theta^T x$, $\theta \in \mathbb{R}^d$ unknown parameter
- **Maximum Information Gain**: $\gamma_T = O(d \ln T)$
- Regret of IGP-UCB is $\tilde{O}(d\sqrt{T})$ and GP-TS is $\tilde{O}(d^{3/2}\sqrt{T})$

- Exactly recovers regrets of OFUL (Abbasi-Yadkori et al., NIPS 2011) and Linear TS (Agrawal and Goyal, ICML 2013)

- Lower Bound: $\Omega(d\sqrt{T})$ (Dani et al., COLT 2008)

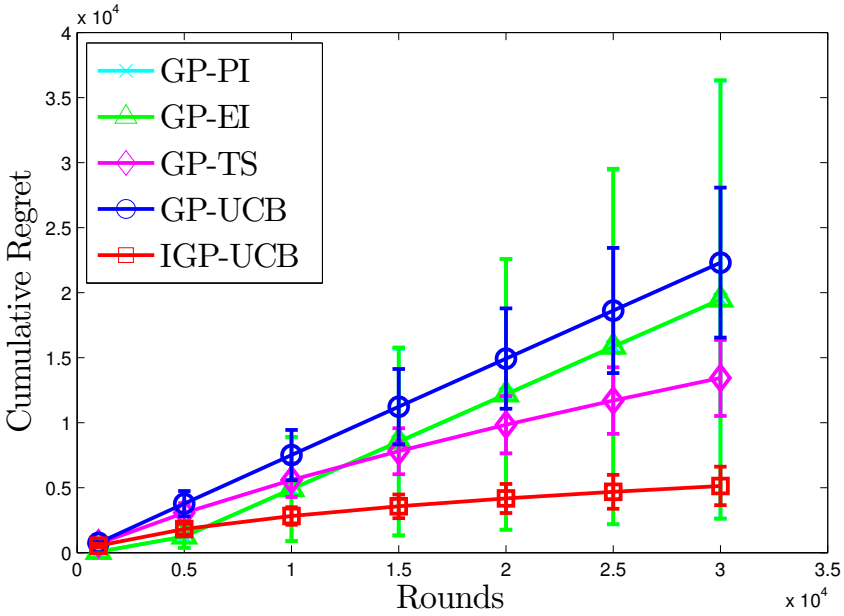
Numerical Results

Algorithms Compared:

1. GP-Expected Improvement (Močkus, 1975)
2. GP-Probabilistic Improvement (Kushner, 1964)
3. GP-UCB (Srinivas et al., 2010)
4. IGP-UCB (this work)
5. GP-TS (this work)

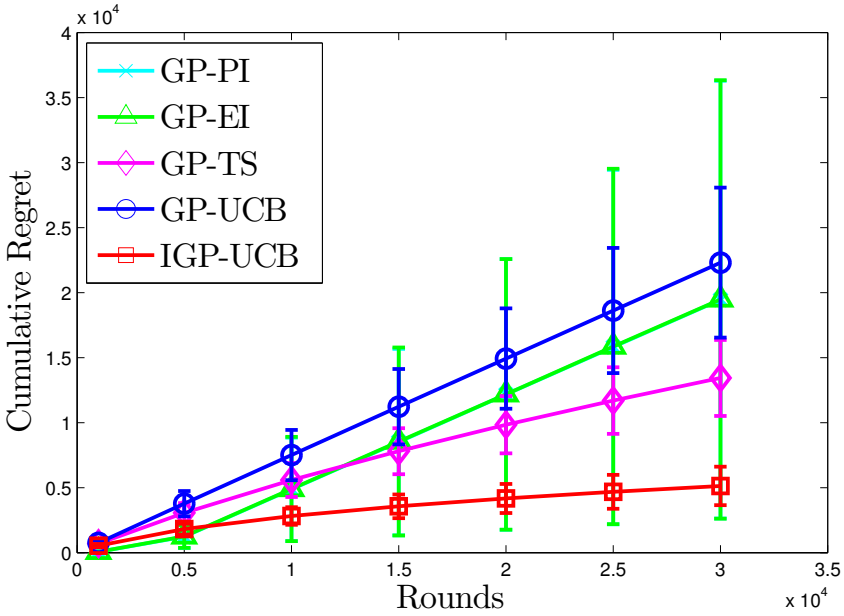
Numerical Results

f sampled from RKHS
(Squared Exponential kernel)



Numerical Results

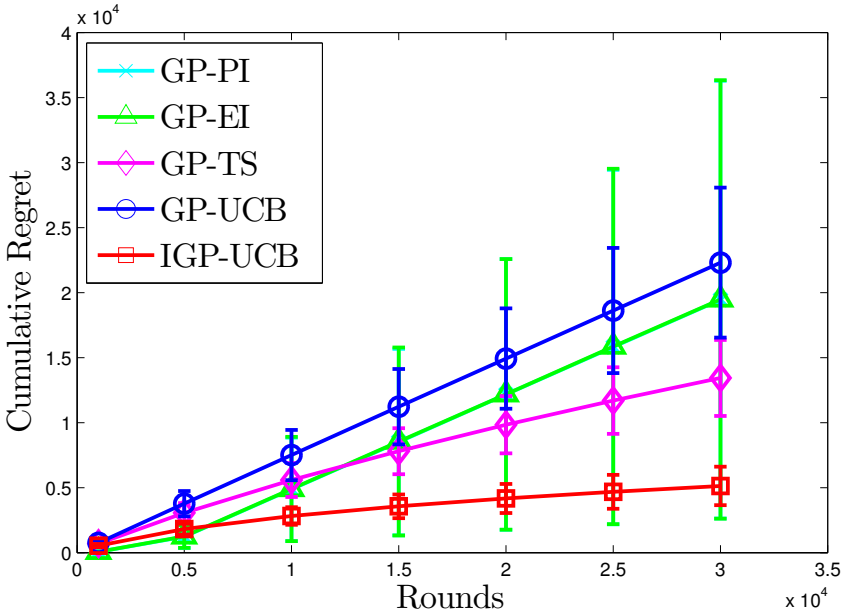
f sampled from RKHS
(Squared Exponential kernel)



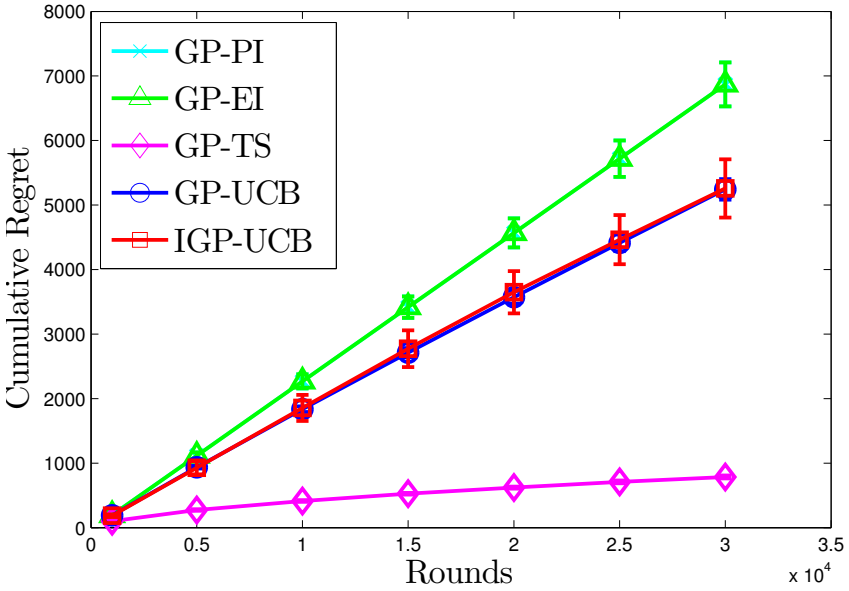
- IGP-UCB improves over GP-UCB ☺☺
- GP-TS fares reasonably well ☺

Numerical Results

f sampled from RKHS
(Squared Exponential kernel)



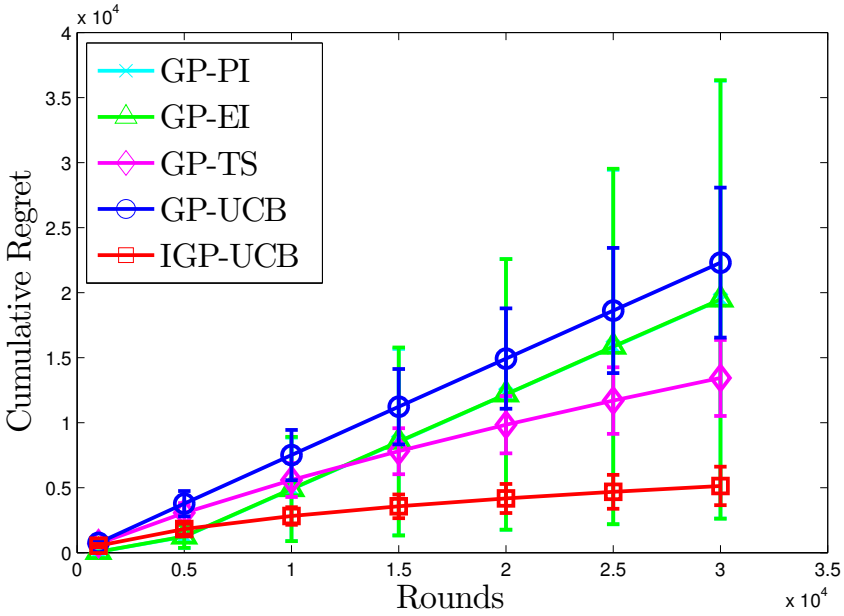
Temperature Sensor Data
(Intel Berkeley Research lab)



- IGP-UCB improves over GP-UCB ☺☺
- GP-TS fares reasonably well ☺

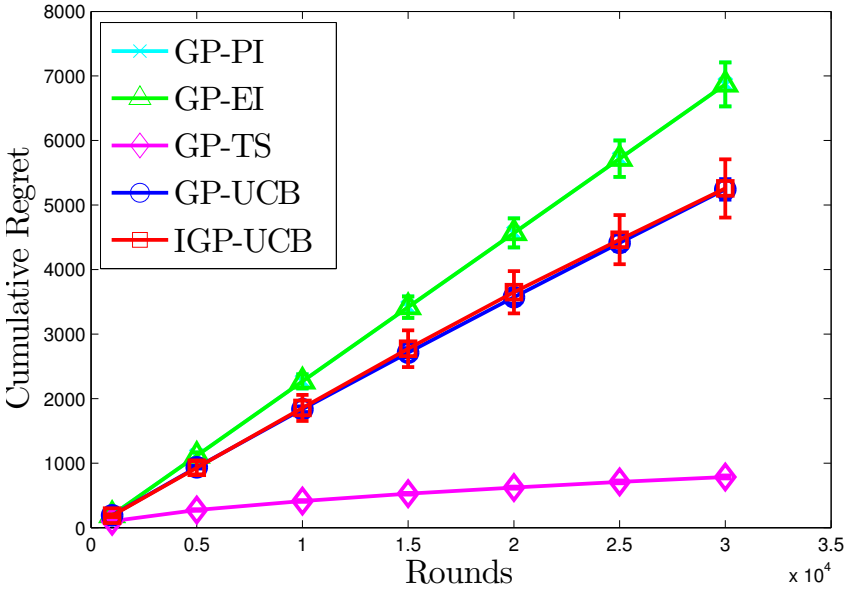
Numerical Results

f sampled from RKHS
(Squared Exponential kernel)



- IGP-UCB improves over GP-UCB ☺☺
- GP-TS fares reasonably well ☺

Temperature Sensor Data
(Intel Berkeley Research lab)



- IGP-UCB performs similar to GP-UCB ✓
- GP-TS performs the best ☺

Key Tool: New Concentration Inequality

Setup:

- Feature map $\varphi : D \to \text{RKHS}$
- $S_t = \sum_{s=1}^{t} \varepsilon_s \varphi(x_s) \leftarrow$ RKHS-valued Martingale
- $V_t = I + \sum_{s=1}^{t} \varphi(x_s)\varphi(x_s)^T \leftarrow$ possibly of infinite dimension

Key Tool: New Concentration Inequality

Setup:

- Feature map $\varphi : D \to \text{RKHS}$
- $S_t = \sum_{s=1}^{t} \varepsilon_s \varphi(x_s) \leftarrow$ RKHS-valued Martingale
- $V_t = I + \sum_{s=1}^{t} \varphi(x_s)\varphi(x_s)^T \leftarrow$ possibly of infinite dimension

Result 3: Self-Normalized CI for RKHS-valued Martingales

- For all t: $\|S_t\|^2_{V_t^{-1}} \leq 2R^2 \ln(\frac{\sqrt{\det(K_t+I)}}{\delta})$ with probability at least $1-\delta$ if K_t is positive-definite
- **Generalizes** finite-dimensional Inequality for vector-valued Martingales (Abbasi-Yadkori et al., NIPS 2011)
- Curse of Dimensionality $\to$ Mixing over Gaussian Processes

Summary

For Non-parametric Bandits :

- **Improved** existing UCB based algorithm
- **Introduced** new Thompson Sampling based algorithm
- **Developed** new self-normalized concentration inequality for RKHS-valued martingales

Summary

For Non-parametric Bandits :

- **Improved** existing UCB based algorithm
- **Introduced** new Thompson Sampling based algorithm
- **Developed** new self-normalized concentration inequality for RKHS-valued martingales

Future Work:

- Kernel function not known to the learner
- Time varying functions from RKHS

Thank You

Poster Tonight