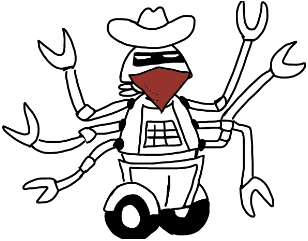
Stochastic Bandits



Kirthevasan Kandasamy

Carnegie Mellon University

University of Moratuwa, Sri Lanka

August 17, 2017

Slides: www.cs.cmu.edu/~kkandasa/misc/mora-slides.pdf

Slides are up on my webpage: www.cs.cmu.edu/~kkandasa

home publications software misc.

kirthevasan kandasamy

PhD Student, Carnegie Mellon University

[CV] [Google Scholar] [GitHub] [Contact]

I am a fourth year Machine Learning PhD student (now ABD) in the School of Computer Science at Carnegie Mellon University. I am co-advised by Jeff Schneider and Barnabas Poczos. I am a member of the Auton Lab and the StatML Group. Prior to CMU, I completed my B.Sc in Electronics & Telecommunications Engineering at the University of Moratuwa, Sri Lanka.

My research interests lie in the intersection of statistical and algorithmic Machine Learning. My current research spans bandit problems, Bayesian optimisation, Gaussian processes, nonparametric statistics and graphical models. As of late, I have also hopped on the deep learning bandwagon.

I am generously supported by a Facebook PhD fellowship (2017) and a CMU Presidential fellowship (2015).

Slides

Preprints

Asynchronous Parallel Bayesian Optimisation via Thompson Sampling
Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, Barnabas Poczos
[arxiv] [AutoML slides] [Moratuwa slides]

Multi-fidelity Gaussian Process Bandit Optimisation
Kirthevasan Kandasamy, Gautam Dasarathy, Junier Oliva, Jeff Schneider, Barnabas Poczos
[arxiv] [code] [UCL slides]

Influence Functions for Machine Learning: Nonparametric Estimators for Entropies, Divergences and Mutual Informations
Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, James Robins

On-line advertising

You are given a pool of 250 ads.

Task:

- You can display one ad at a time, (say for 10^6 times).
- You wish to maximise the cumulative number of clicks, i.e. identify ads with the highest click-through-rate and display them most of the time.

The Stochastic Multi-armed Bandit (Robbins, 1952)

- You are given K arms, $\mathcal{X} = \{1, \ldots, K\}$.
- At every round you "play/pull" an arm.

The Stochastic Multi-armed Bandit (Robbins, 1952)

- You are given K arms, $\mathcal{X} = \{1, \ldots, K\}$.
- At every round you "play/pull" an arm.
- When you play arm $x_t \in \mathcal{X}$ in round t you receive a *stochastic* reward y_t, where $\mathbb{E}[y_t] = f(x_t)$.

The Stochastic Multi-armed Bandit (Robbins, 1952)

- You are given K arms, $\mathcal{X} = \{1, \ldots, K\}$.
- At every round you "play/pull" an arm.
- When you play arm $x_t \in \mathcal{X}$ in round t you receive a *stochastic* reward y_t, where $\mathbb{E}[y_t] = f(x_t)$.
- **Goal:** Maximise the cumulative sum of expected rewards,

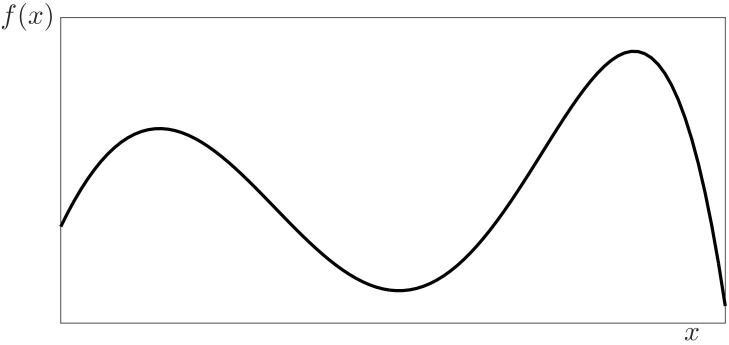
$$\mathbb{E}\left[\sum_{t=1}^{n} y_t\right] = \sum_{t=1}^{n} f(x_t).$$

The Stochastic Multi-armed Bandit (Robbins, 1952)

- You are given K arms, $\mathcal{X} = \{1, \ldots, K\}$.
- At every round you "play/pull" an arm.
- When you play arm $x_t \in \mathcal{X}$ in round t you receive a *stochastic* reward y_t, where $\mathbb{E}[y_t] = f(x_t)$.
- **Goal:** Maximise the cumulative sum of expected rewards,
$$\mathbb{E}\left[\sum_{t=1}^{n} y_t\right] = \sum_{t=1}^{n} f(x_t).$$
- **Goal:** An algorithm (policy/strategy) which achieves "small" *cumulative regret*,
$$R_n = \sum_{t=1}^{n} f(x_\star) - \sum_{t=1}^{n} f(x_t) = \sum_{t=1}^{n} \left(f(x_\star) - f(x_t)\right).$$
where, $x_\star = \mathrm{argmax}_{x\in\mathcal{X}} f(x)$.

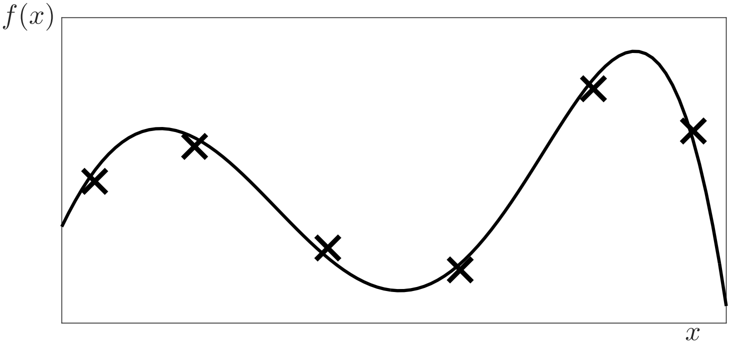
Smooth Bandits

$f : \mathcal{X} \to \mathbb{R}$ is a black-box function that is accessible only via noisy evaluations. $\mathcal{X}$ is a metric space, e.g. $\mathbb{R}^d$.



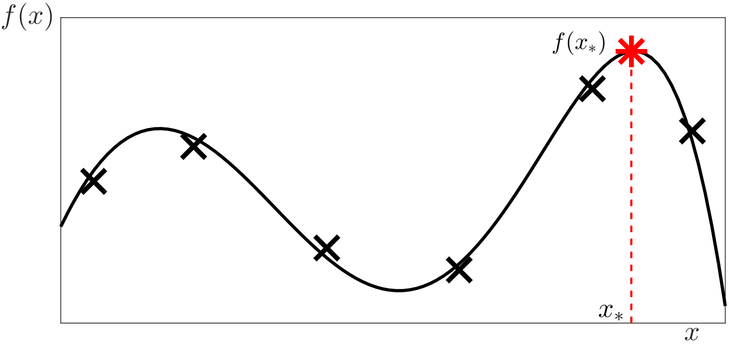
Smooth Bandits

$f : \mathcal{X} \to \mathbb{R}$ is a black-box function that is accessible only via noisy evaluations. $\mathcal{X}$ is a metric space, e.g. $\mathbb{R}^d$.



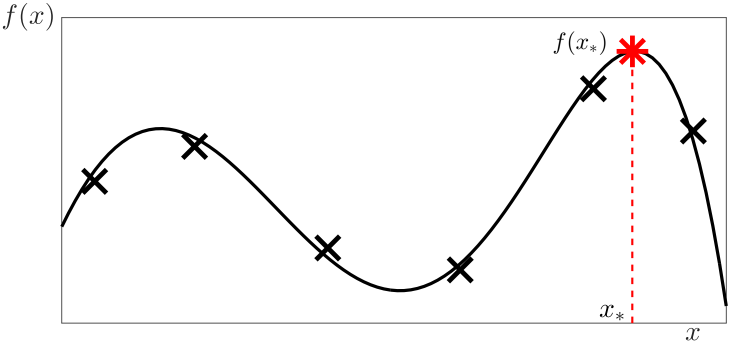
Smooth Bandits

$f : \mathcal{X} \to \mathbb{R}$ is a black-box function that is accessible only via noisy evaluations. $\mathcal{X}$ is a metric space, e.g. $\mathbb{R}^d$.
Let $x_\star = \mathrm{argmax}_x f(x)$.



Smooth Bandits

$f : \mathcal{X} \to \mathbb{R}$ is a black-box function that is accessible only via noisy evaluations. $\mathcal{X}$ is a metric space, e.g. $\mathbb{R}^d$.
Let $x_\star = \mathrm{argmax}_x f(x)$.

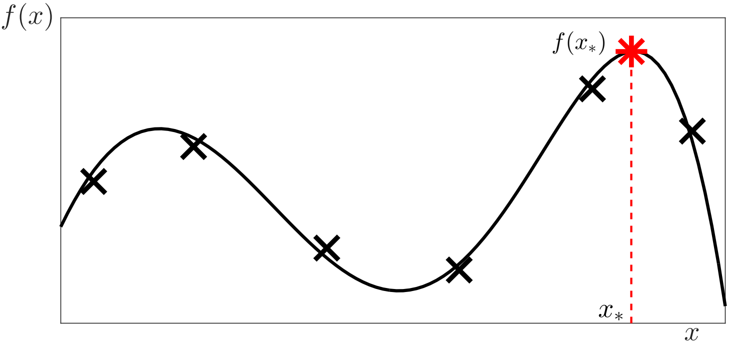


Cumulative Regret after n evaluations

$$R_n = \sum_{t=1}^{n} f(x_\star) - f(x_t).$$

Smooth Bandits

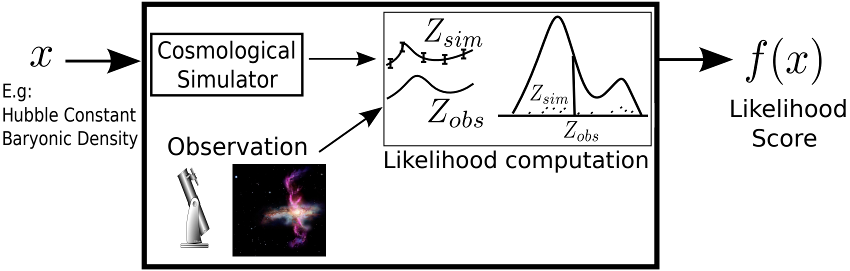
$f : \mathcal{X} \to \mathbb{R}$ is a black-box function that is accessible only via noisy evaluations. $\mathcal{X}$ is a metric space, e.g. $\mathbb{R}^d$.
Let $x_\star = \mathrm{argmax}_x\, f(x)$.

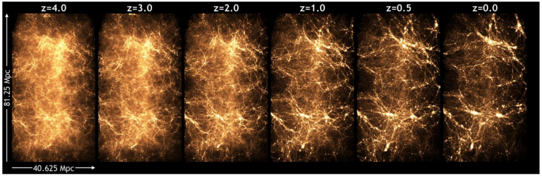


Simple Regret after n evaluations

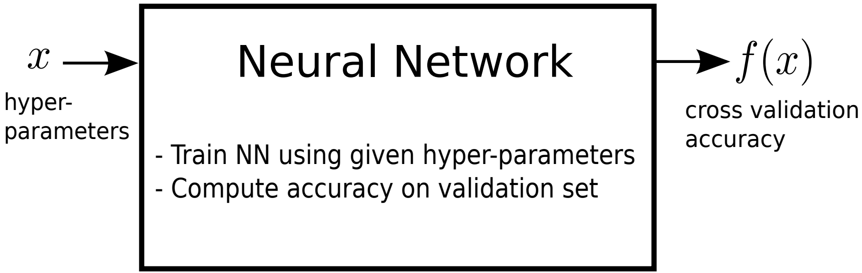
$$S_n = f(x_\star) \; - \max_{t=1,\ldots,n} f(x_t).$$

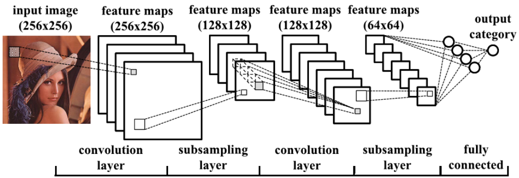
Applications





Applications

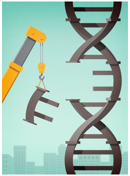




Applications

$x \longrightarrow$ **Expensive Blackbox Function** $\longrightarrow f(x)$

Other Examples:
- Pre-clinical Drug Discovery
- Optimal policy in Autonomous Driving
- Synthetic gene design



Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

$f : \mathcal{X} \to \mathbb{R}$. On playing $x \in \mathcal{X}$ you observe $f(x) + \varepsilon$, $\mathbb{E}\varepsilon = 0$.

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

$f : \mathcal{X} \to \mathbb{R}$. On playing $x \in \mathcal{X}$ you observe $f(x) + \varepsilon$, $\mathbb{E}\varepsilon = 0$.

Two notions of regret

1. Cumulative regret, $R_n = \sum_{t=1}^{n} f(x_\star) - f(x_t)$.
2. Simple regret, $S_n = f(x_\star) - \max_{t=1,\dots,n} f(x_t)$.

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

$f : \mathcal{X} \to \mathbb{R}$. On playing $x \in \mathcal{X}$ you observe $f(x) + \varepsilon$, $\mathbb{E}\varepsilon = 0$.

Two notions of regret

1. Cumulative regret, $R_n = \sum_{t=1}^{n} f(x_\star) - f(x_t)$.
2. Simple regret, $S_n = f(x_\star) - \max_{t=1,\dots,n} f(x_t)$.

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

$f : \mathcal{X} \to \mathbb{R}$. On playing $x \in \mathcal{X}$ you observe $f(x) + \varepsilon$, $\mathbb{E}\varepsilon = 0$.

Two notions of regret

1. Cumulative regret, $R_n = \sum_{t=1}^{n} f(x_\star) - f(x_t)$.
2. Simple regret, $S_n = f(x_\star) - \max_{t=1,\ldots,n} f(x_t)$.

Other formalisms: contextual bandit, adversarial bandit, duelling bandit, linear bandit, best arm identification and several more ...

Recap

Types of arms (domain $\mathcal{X}$)

1. K-armed bandit, $\mathcal{X}$ is a finite set.
2. Smooth bandit, $\mathcal{X}$ is a metric space (e.g. $\mathbb{R}^d$).
3. "Smooth K-armed" bandits, $\mathcal{X}$ is finite, but there is additional structure.

$f : \mathcal{X} \to \mathbb{R}$. On playing $x \in \mathcal{X}$ you observe $f(x) + \varepsilon$, $\mathbb{E}\varepsilon = 0$.

Two notions of regret

1. Cumulative regret, $R_n = \sum_{t=1}^{n} f(x_\star) - f(x_t)$.
2. Simple regret, $S_n = f(x_\star) - \max_{t=1,\dots,n} f(x_t)$.

Other formalisms: contextual bandit, adversarial bandit, duelling bandit, linear bandit, best arm identification and several more ...

N.B: Pulling/playing an arm = experiment = function evaluation

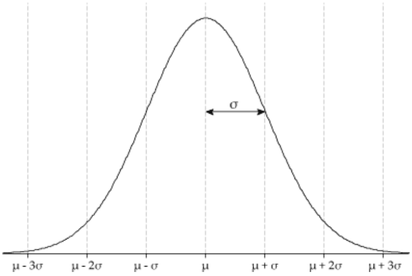
Outline

- Part I: Stochastic bandits (cont'd)
    1. Gaussian processes for smooth bandits
    2. Algorithms: Upper Confidence Bound (UCB) & Thompson Sampling (TS)
- Digression: SL2College Research Collaboration Program
- Part II: My research
    1. Multi-fidelity bandit: cheap approximations to an expensive experiments
    2. Parallelising arm pulls

Outline

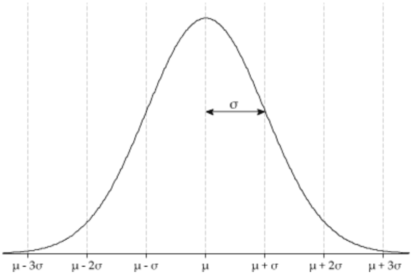
- Part I: Stochastic bandits (cont'd)
  1. Gaussian processes for smooth bandits
  2. Algorithms: Upper Confidence Bound (UCB) & Thompson Sampling (TS)
- Digression: SL2College Research Collaboration Program
- Part II: My research
  1. Multi-fidelity bandit: cheap approximations to an expensive experiments
  2. Parallelising arm pulls

Gaussian (Normal) distribution $\mathcal{N}(\mu, \sigma^2)$



- A probability distribution for real valued random variables.
- Mean μ and variance σ^2 completely characterises distribution.

Gaussian (Normal) distribution $\mathcal{N}(\mu, \sigma^2)$



- A probability distribution for real valued random variables.
- Mean μ and variance σ^2 completely characterises distribution.
- For samples $X_1, \ldots, X_n$, let $\hat{\mu} = \frac{1}{n}\sum_i X_i$ be the sample mean. Then, $\hat{\mu} \pm 1.96\frac{\sigma}{\sqrt{n}}$ is a 95% confidence interval for μ.
- Can draw samples (e.g. in Matlab: `mu + sigma * randn()`).

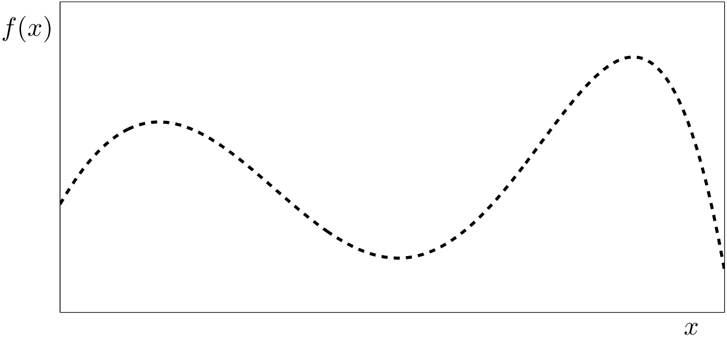
Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

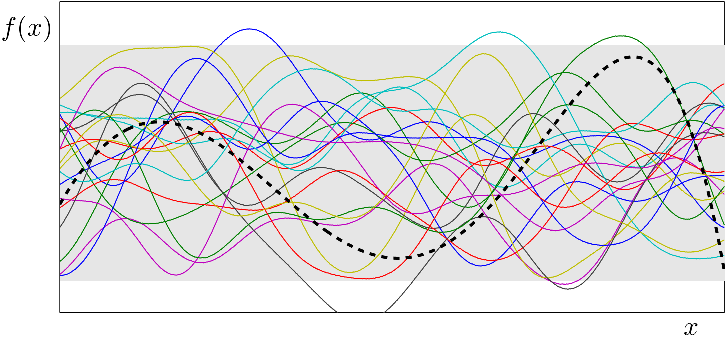
Functions with no observations



Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

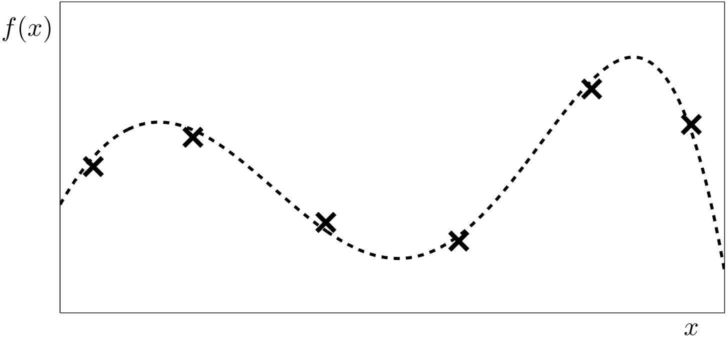
Prior $\mathcal{GP}$



Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

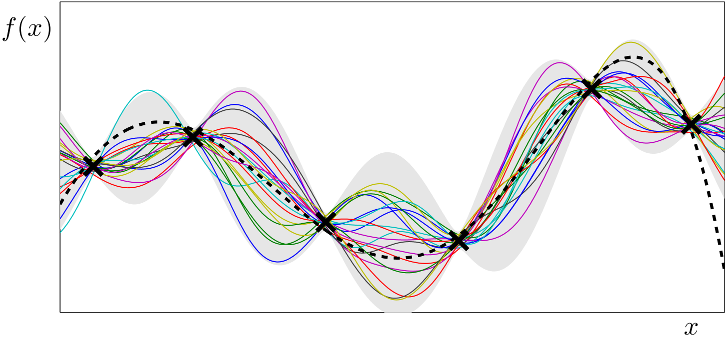
Observations



Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

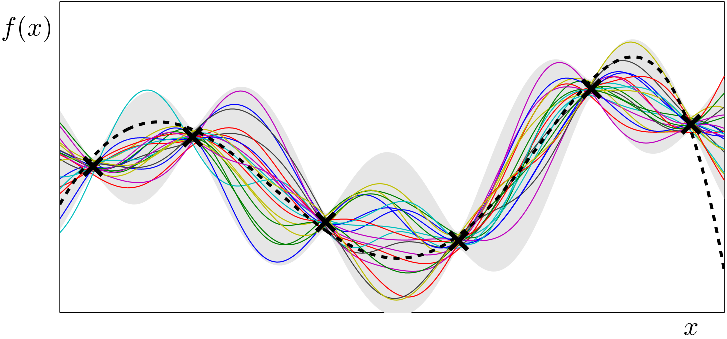
Posterior $\mathcal{GP}$ given observations



Gaussian Processes ($\mathcal{GP}$)

$\mathcal{GP}(\mu, \kappa)$: A distribution over functions from $\mathcal{X}$ to $\mathbb{R}$.
Completely characterised by mean function $\mu : \mathcal{X} \to \mathbb{R}$, and covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Posterior $\mathcal{GP}$ given observations



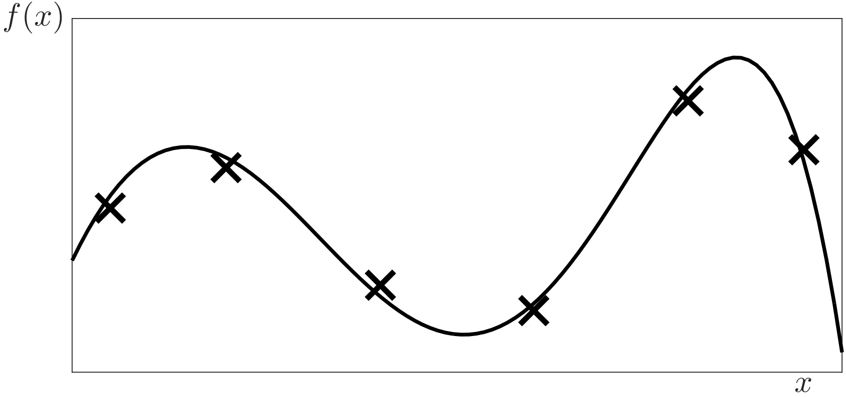
After t observations, $\quad f(x) \sim \mathcal{N}(\,\mu_t(x),\, \sigma_t^2(x)\,)$.

Algorithm 1: Upper Confidence Bounds in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010).

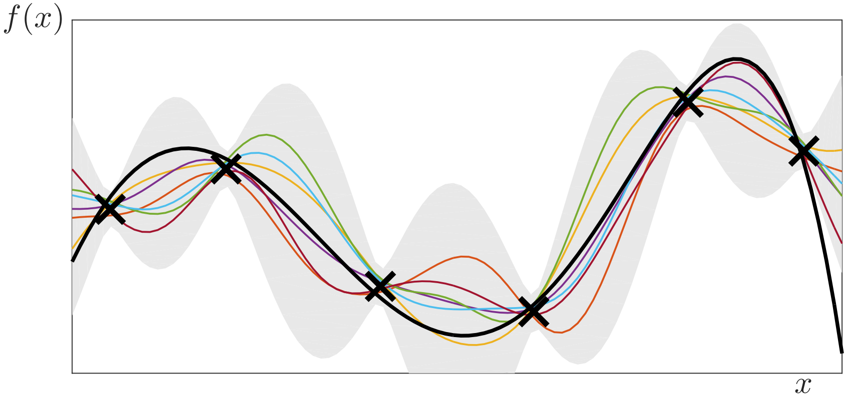


Algorithm 1: Upper Confidence Bounds in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010).

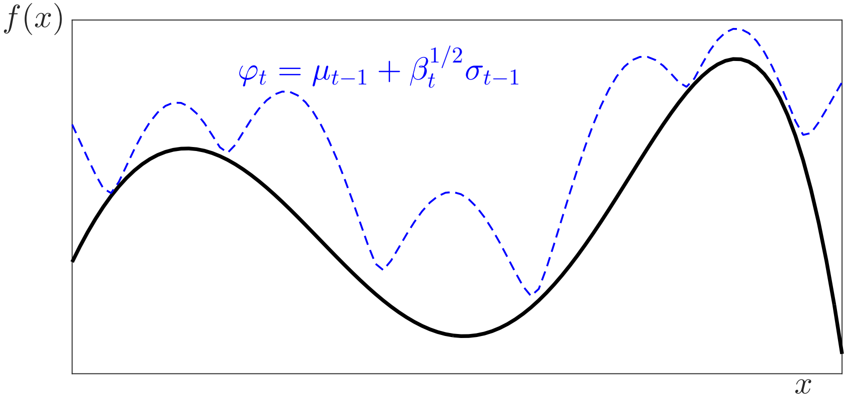


Algorithm 1: Upper Confidence Bounds in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010).



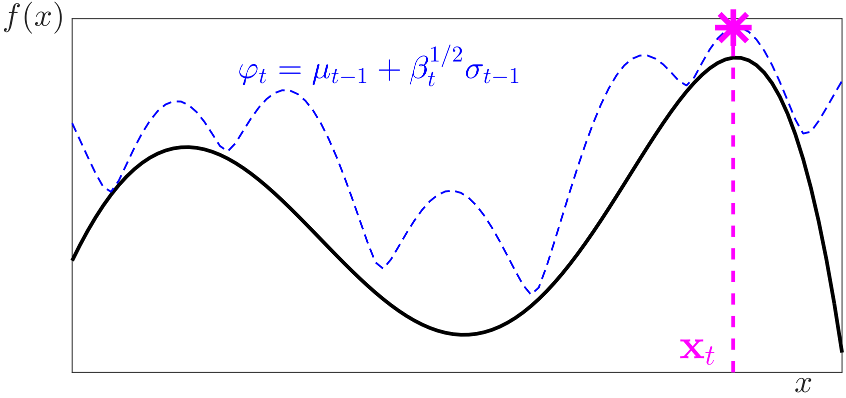
Construct upper conf. bound: $\varphi_t(x) = \mu_{t-1}(x) + \beta_t^{1/2}\sigma_{t-1}(x)$.

Algorithm 1: Upper Confidence Bounds in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010).



Maximise upper confidence bound.

GP-UCB

$$x_t = \underset{x}{\operatorname{argmax}} \;\; \mu_{t-1}(x) + \beta_t^{1/2}\sigma_{t-1}(x)$$

- μ_{t-1}: Exploitation
- σ_{t-1}: Exploration
- β_t controls the tradeoff. $\beta_t \asymp \log t$.

GP-UCB

$$x_t = \operatorname*{argmax}_x \;\; \mu_{t-1}(x) + \beta_t^{1/2}\sigma_{t-1}(x)$$

- μ_{t-1}: Exploitation
- σ_{t-1}: Exploration
- β_t controls the tradeoff. $\beta_t \asymp \log t$.

GP-UCB, κ is an SE kernel (Srinivas et al. 2010)

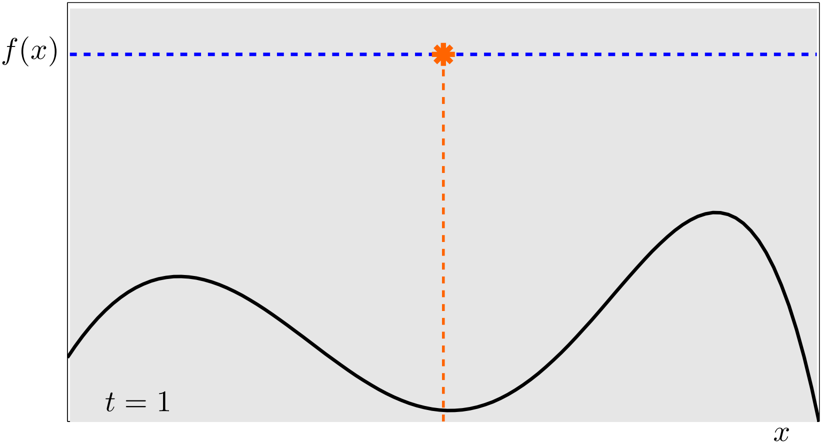
$$\text{w.h.p} \qquad S_n \;=\; f(x_\star) - \max_{t=1,\dots,n} f(x_t) \;\lesssim\; \sqrt{\frac{\log(n)^d \mathrm{vol}(\mathcal{X})}{n}}$$

GP-UCB

(Srinivas et al. 2010)

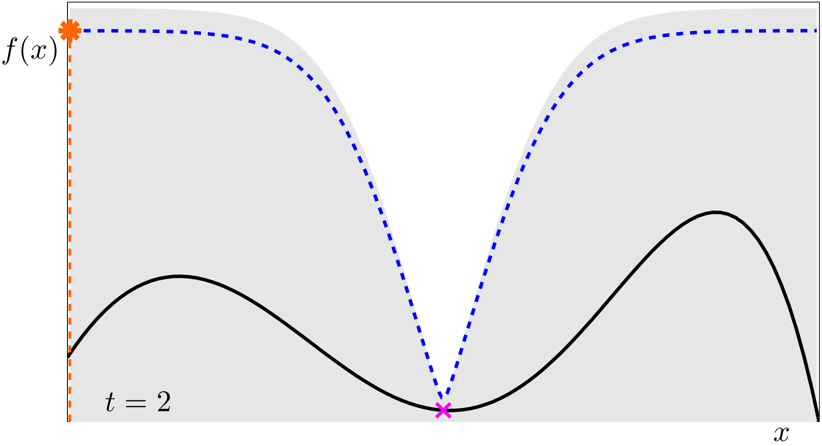
GP-UCB

(Srinivas et al. 2010)



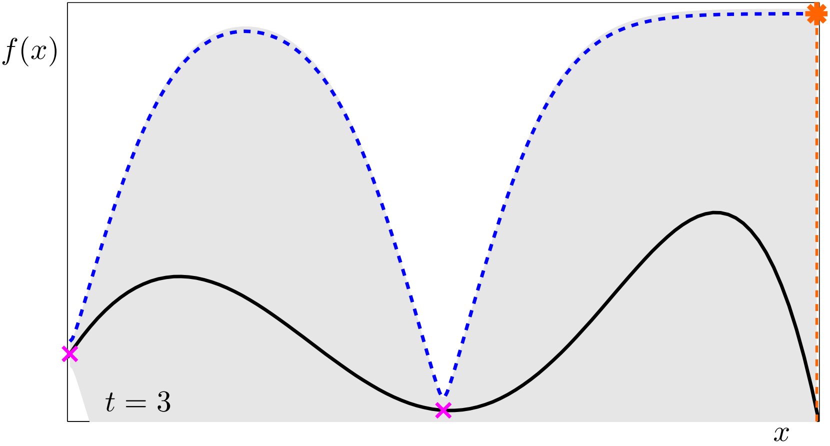
GP-UCB

(Srinivas et al. 2010)



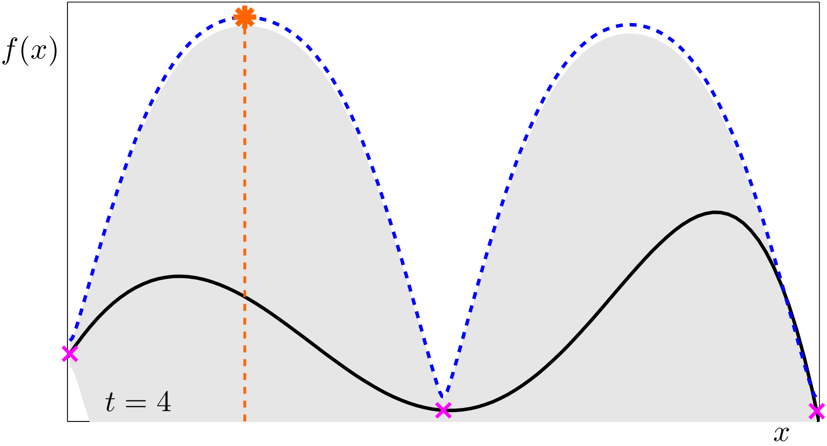
GP-UCB

(Srinivas et al. 2010)



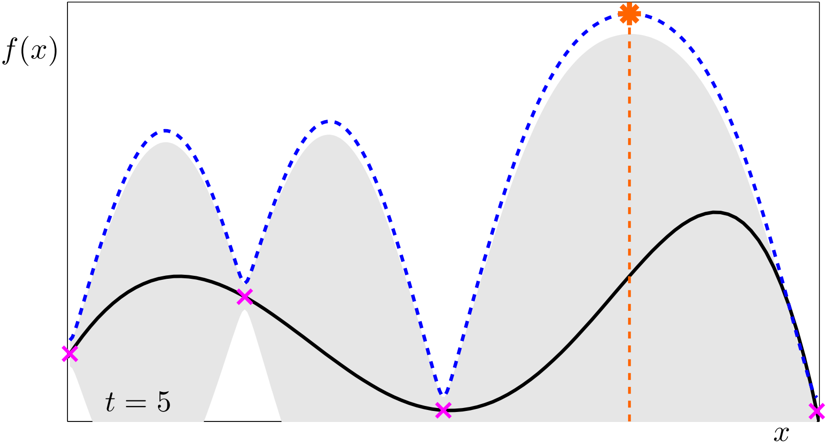
GP-UCB

(Srinivas et al. 2010)



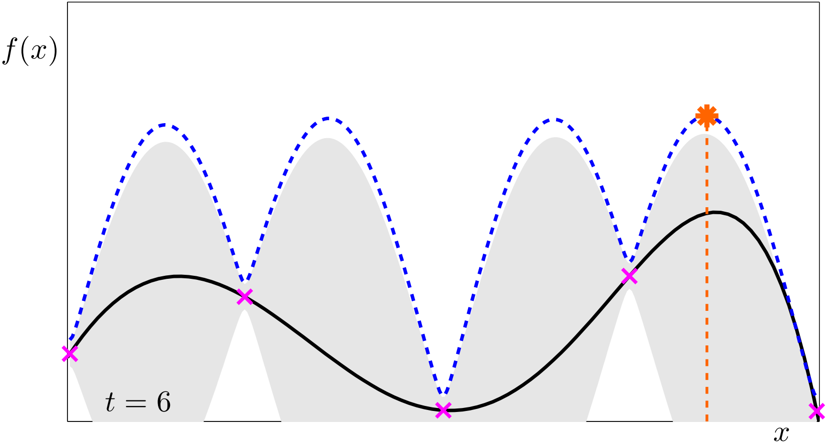
GP-UCB

(Srinivas et al. 2010)



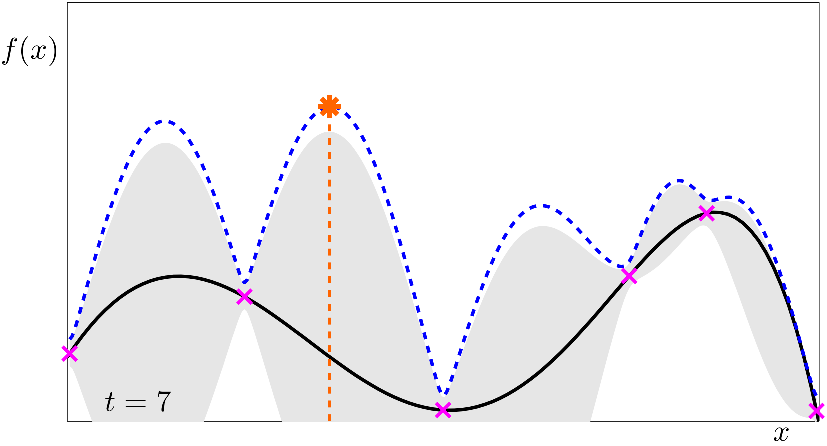
GP-UCB

(Srinivas et al. 2010)



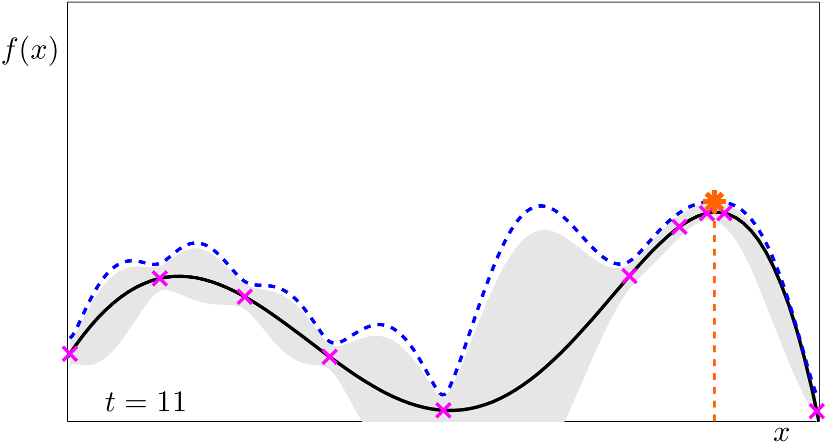
GP-UCB

(Srinivas et al. 2010)



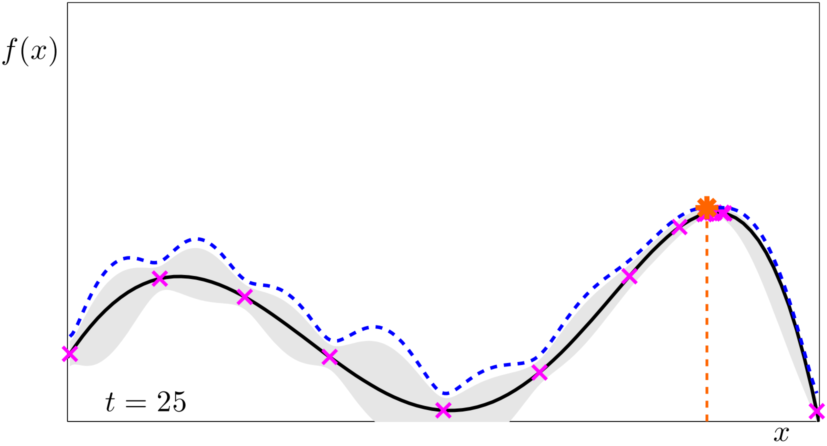
GP-UCB

(Srinivas et al. 2010)



GP-UCB

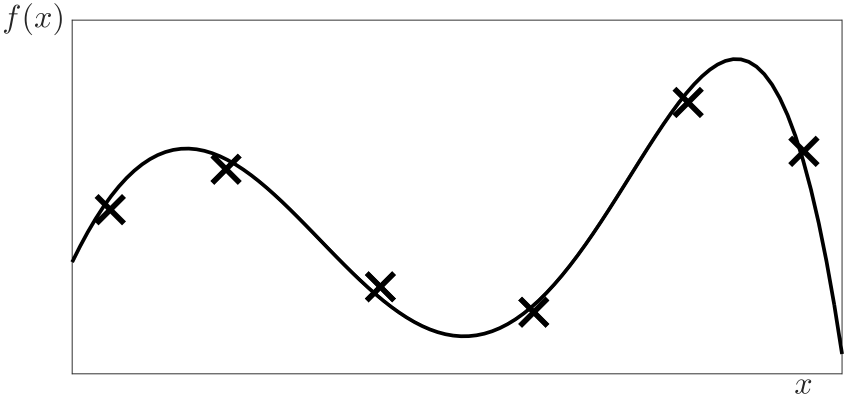
(Srinivas et al. 2010)



Algorithm 2: Thompson Sampling in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

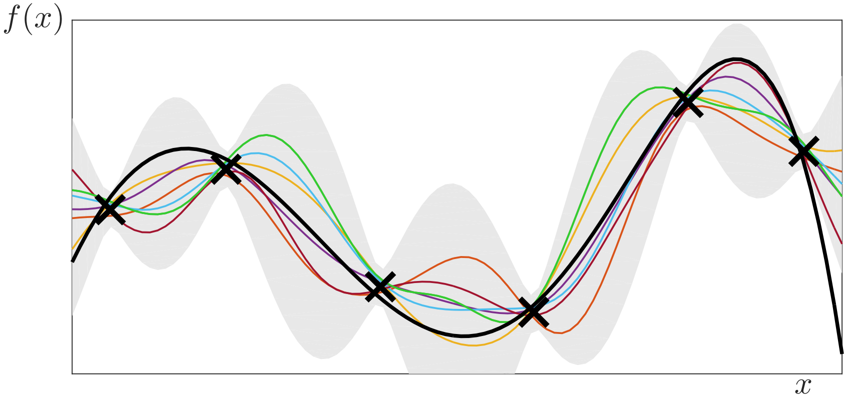
Thompson Sampling (TS) (Thompson, 1933).



Algorithm 2: Thompson Sampling in GP Bandits

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

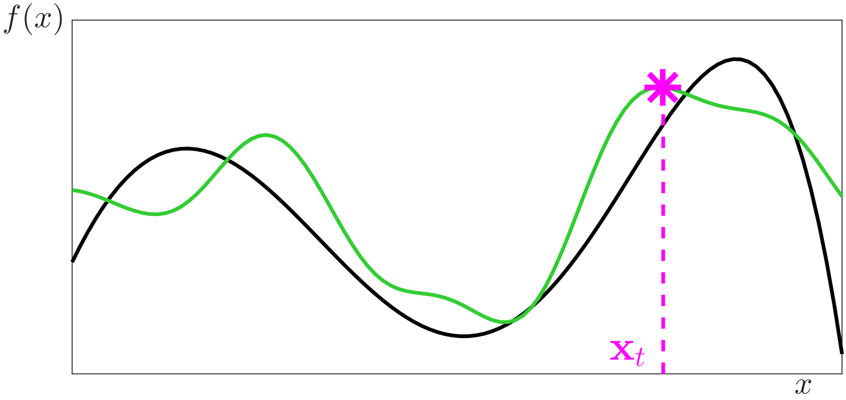
Thompson Sampling (TS) (Thompson, 1933).



Algorithm 2: Thompson Sampling in GP Bandits

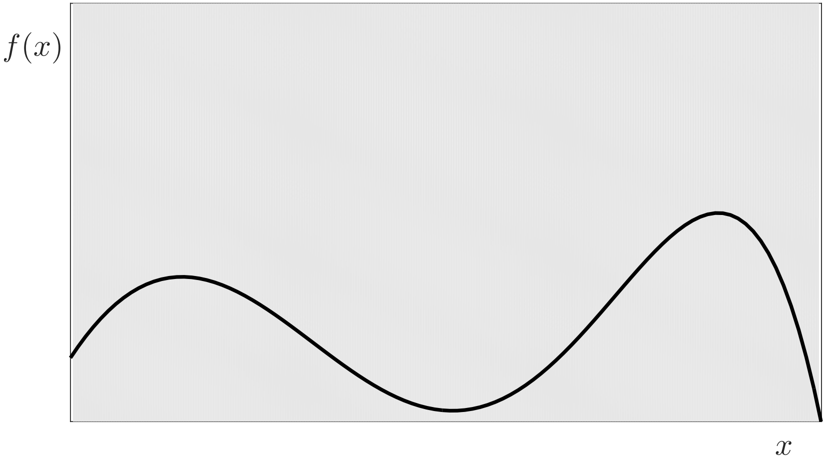
Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Thompson Sampling (TS) (Thompson, 1933).

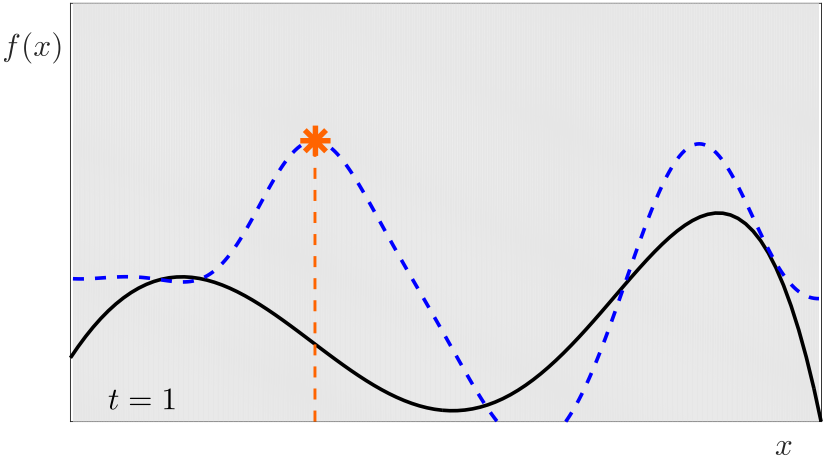


Draw sample g from posterior. Choose $x_t = \mathrm{argmax}_x\, g(x)$.

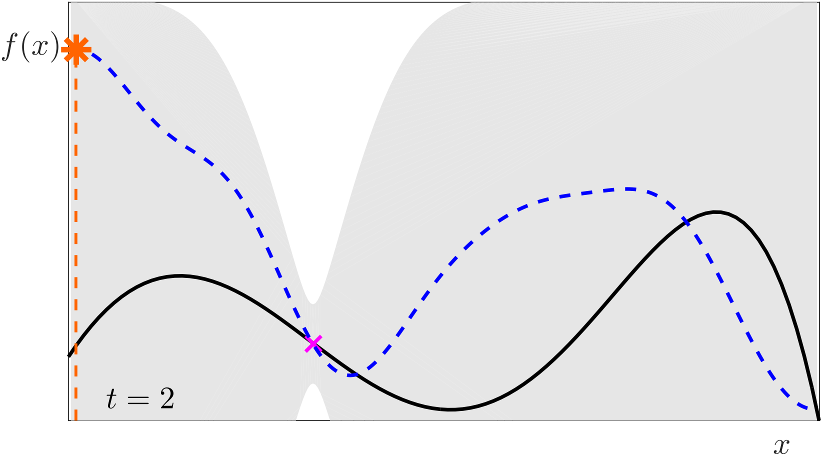
Thompson Sampling (TS) in GPs (Thompson, 1933)



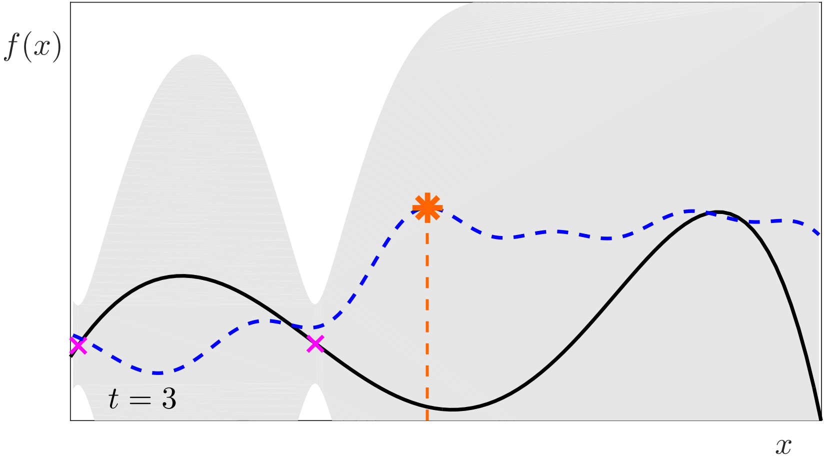
Thompson Sampling (TS) in GPs (Thompson, 1933)



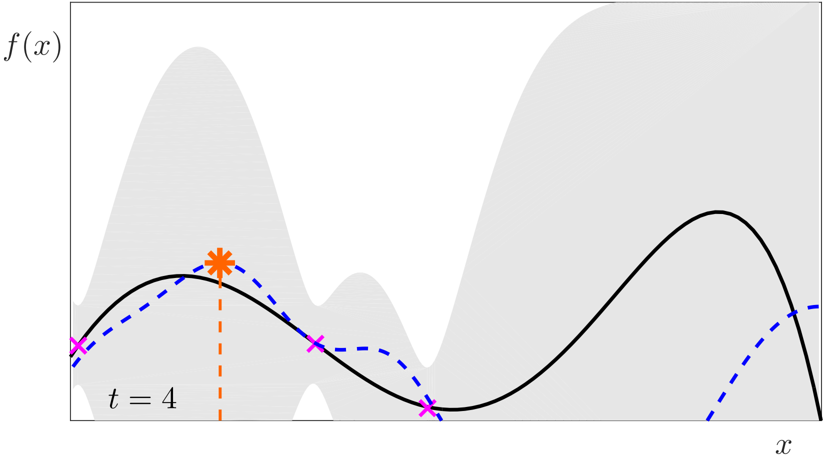
Thompson Sampling (TS) in GPs (Thompson, 1933)



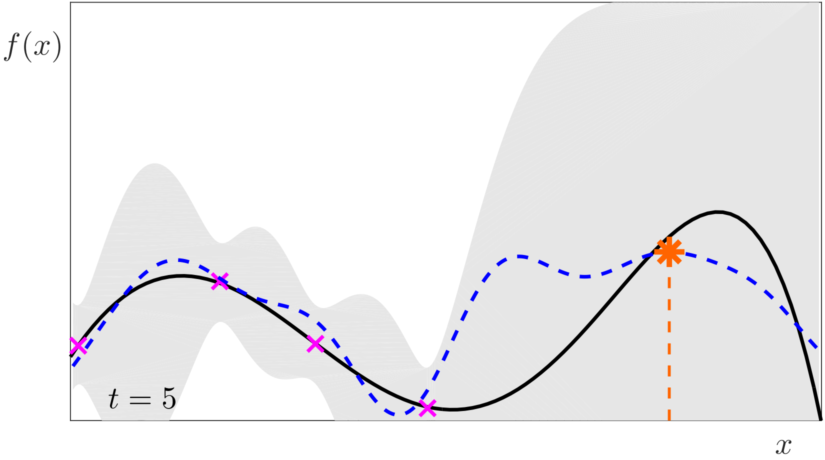
Thompson Sampling (TS) in GPs (Thompson, 1933)



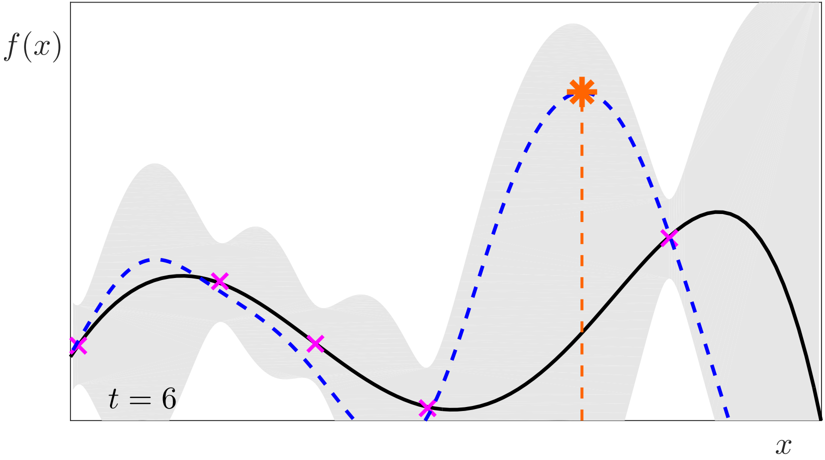
Thompson Sampling (TS) in GPs (Thompson, 1933)



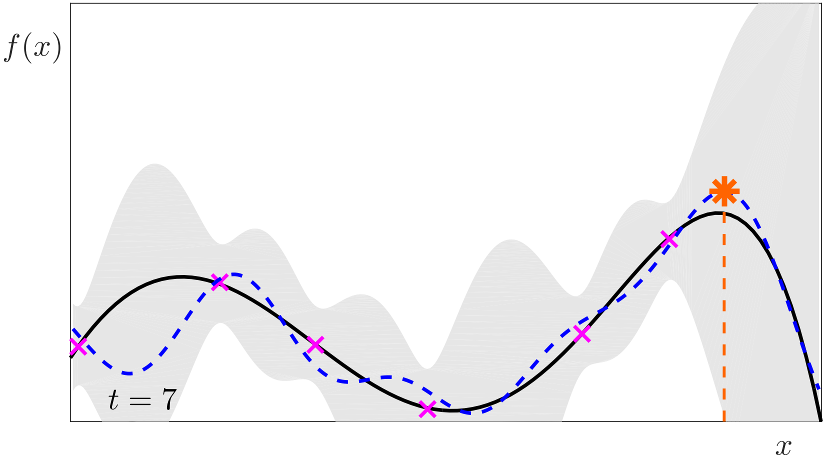
Thompson Sampling (TS) in GPs (Thompson, 1933)



Thompson Sampling (TS) in GPs (Thompson, 1933)

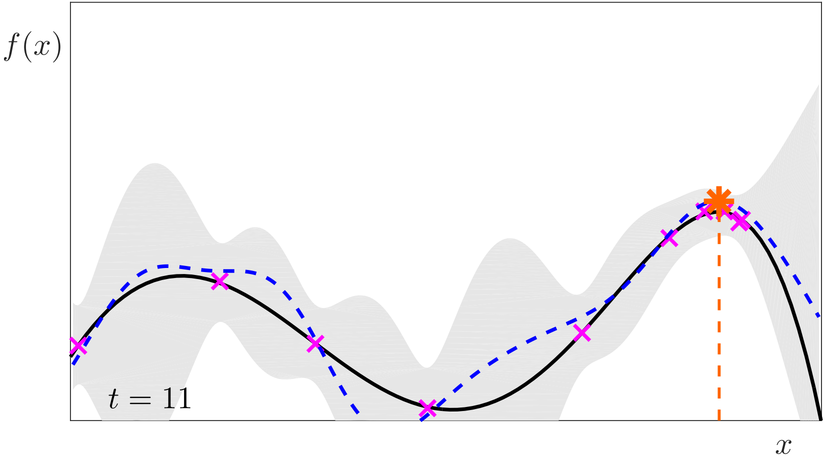


Thompson Sampling (TS) in GPs (Thompson, 1933)

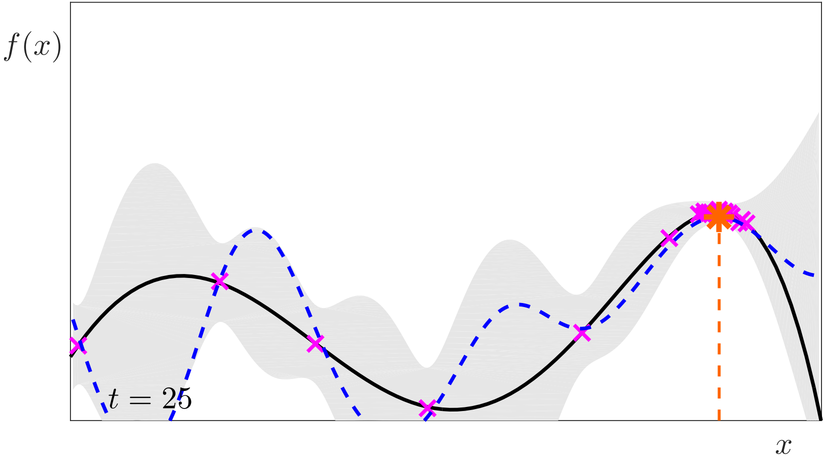


Thompson Sampling (TS) in GPs

(Thompson, 1933)



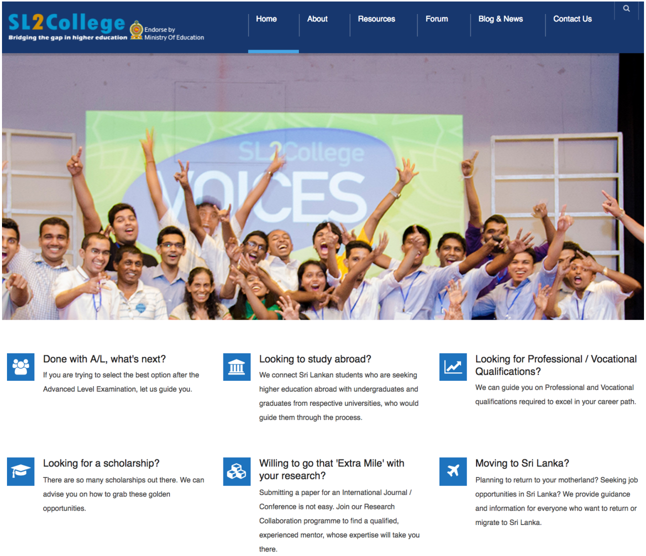
Thompson Sampling (TS) in GPs (Thompson, 1933)



Outline

- Part I: Stochastic bandits (cont'd)
  1. Gaussian processes for smooth bandits
  2. Algorithms: Upper Confidence Bound (UCB) & Thompson Sampling (TS)
- Digression: SL2College Research Collaboration Program
- Part II: My research
  1. Multi-fidelity bandit: cheap approximations to an expensive experiments
  2. Parallelising arm pulls

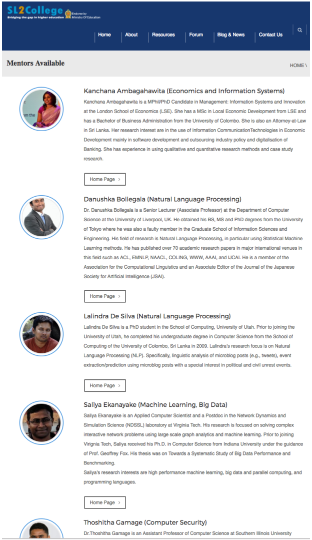
SL2College



www.sl2college.org

SL2College Research Collaboration Program

-Ashwin de Silva



SL2College Bridging the gap in higher education Endorse by Ministry Of Education

Home | About | Resources | Forum | Blog & News | Contact Us

Mentors Available

HOME \

Kanchana Ambagahawita (Economics and Information Systems)

Kanchana Ambagahawita is a MPhil/PhD Candidate in Management: Information Systems and Innovation at the London School of Economics (LSE). She has a MSc in Local Economic Development from LSE and has a Bachelor of Business Administration from the University of Colombo. She is also an Attorney-at-Law in Sri Lanka. Her research interest are in the use of Information CommunicationTechnologies in Economic Development mainly in software development and outsourcing industry policy and digitalisation of Banking. She has experience in using qualitative and quantitative research methods and case study research.

Home Page ›

Danushka Bollegala (Natural Language Processing)

Dr. Danushka Bollegala is a Senior Lecturer (Associate Professor) at the Department of Computer Science at the University of Liverpool, UK. He obtained his BS, MS and PhD degrees from the University of Tokyo where he was also a faulty member in the Graduate School of Information Sciences and Engineering. His field of research is Natural Language Processing, in particular using Statistical Machine Learning methods. He has published over 70 academic research papers in major international venues in this field such as ACL, EMNLP, NAACL, COLING, WWW, AAAI, and IJCAI. He is a member of the Association for the Computational Linguistics and an Associate Editor of the Journal of the Japanese Society for Artificial Intelligence (JSAI).

Home Page ›

Lalindra De Silva (Natural Language Processing)

Lalindra De Silva is a PhD student in the School of Computing, University of Utah. Prior to joining the University of Utah, he completed his undergraduate degree in Computer Science from the School of Computing of the University of Colombo, Sri Lanka in 2009. Lalindra's research focus is on Natural Language Processing (NLP). Specifically, linguistic analysis of microblog posts (e.g., tweets), event extraction/prediction using microblog posts with a special interest in political and civil unrest events.

Home Page ›

Saliya Ekanayake (Machine Learning, Big Data)

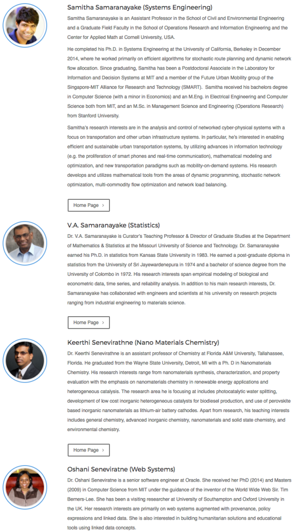
Saliya Ekanayake is an Applied Computer Scientist and a Postdoc in the Network Dynamics and Simulation Science (NDSSL) laboratory at Virginia Tech. His research is focused on solving complex interactive network problems using large scale graph analytics and machine learning. Prior to joining Virignia Tech, Saliya received his Ph.D. in Computer Science from Indiana University under the guidance of Prof. Geoffrey Fox. His thesis was on Towards a Systematic Study of Big Data Performance and Benchmarking.

Saliya's research interests are high performance machine learning, big data and parallel computing, and programming languages.

Home Page ›

Thoshitha Gamage (Computer Security)

Dr.Thoshitha Gamage is an Assistant Professor of Computer Science at Southern Illinois University



Samitha Samaranayake (Systems Engineering)

Samitha Samaranayake is an Assistant Professor in the School of Civil and Environmental Engineering and a Graduate Field Faculty in the School of Operations Research and Information Engineering and the Center for Applied Math at Cornell University, USA.

He completed his Ph.D. in Systems Engineering at the University of California, Berkeley in December 2014, where he worked primarily on efficient algorithms for stochastic route planning and dynamic network flow allocation. Since graduating, Samitha has been a Postdoctoral Associate in the Laboratory for Information and Decision Systems at MIT and a member of the Future Urban Mobility group of the Singapore-MIT Alliance for Research and Technology (SMART). Samitha received his bachelors degree in Computer Science (with a minor in Economics) and an M.Eng. in Electrical Engineering and Computer Science both from MIT, and an M.Sc. in Management Science and Engineering (Operations Research) from Stanford University.

Samitha's research interests are in the analysis and control of networked cyber-physical systems with a focus on transportation and other urban infrastructure systems. In particular, he's interested in enabling efficient and sustainable urban transportation systems, by utilizing advances in information technology (e.g. the proliferation of smart phones and real-time communication), mathematical modeling and optimization, and new transportation paradigms such as mobility-on-demand systems. His research develops and utilizes mathematical tools from the areas of dynamic programming, stochastic network optimization, multi-commodity flow optimization and network load balancing.

Home Page ›

V.A. Samaranayake (Statistics)

Dr. V.A. Samaranayake is Curator's Teaching Professor & Director of Graduate Studies at the Department of Mathematics & Statistics at the Missouri University of Science and Technology. Dr. Samaranayake earned his Ph.D. in statistics from Kansas State University in 1983. He earned a post-graduate diploma in statistics from the University of Sri Jayewardenepura in 1974 and a bachelor of science degree from the University of Colombo in 1972. His research interests span empirical modeling of biological and econometric data, time series, and reliability analysis. In addition to his main research interests, Dr. Samaranayake has collaborated with engineers and scientists at his university on research projects ranging from industrial engineering to materials science.

Home Page ›

Keerthi Senevirathne (Nano Materials Chemistry)

Dr. Keerthi Senevirathne is an assistant professor of Chemistry at Florida A&M University, Tallahassee, Florida. He graduated from the Wayne State University, Detroit, MI with a Ph. D in Nanomaterials Chemistry. His research interests range from nanomaterials synthesis, characterization, and property evaluation with the emphasis on nanomaterials chemistry in renewable energy applications and heterogeneous catalysis. The research area he is focusing at includes photocatalytic water splitting, development of low cost inorganic heterogeneous catalysts for biodiesel production, and use of perovskite based inorganic nanomaterials as lithium-air battery cathodes. Apart from research, his teaching interests includes general chemistry, advanced inorganic chemistry, nanomaterials and solid state chemistry, and environmental chemistry.

Home Page ›

Oshani Seneviratne (Web Systems)

Dr. Oshani Seneviratne is a senior software engineer at Oracle. She received her PhD (2014) and Masters (2009) in Computer Science from MIT under the guidance of the inventor of the World Wide Web Sir. Tim Berners-Lee. She has been a visiting researcher at University of Southampton and Oxford University in the UK. Her research interests are primarily on web systems augmented with provenance, policy expressions and linked data. She is also interested in building humanitarian solutions and educational tools using linked data concepts.

www.sl2college.org/research-collab

research-collab@sl2college.org

SL2College Research Collaboration Program

How it works

We have a pool of doctoral/post-doctoral/professorial mentors (all Sri Lankan).

We connect Sri Lankan undergrads to mentors, who will guide the students on a research project.

Aim: Publish a paper (at a good venue) within a 9-15 month time frame.

Application Process

- Fill out the application form on our webpage: `www.sl2college.org/research-collab`
  - mention areas of interests and preferred mentors.
- .. and email your CV to `research-collab@sl2college.org`.

Application Process

- Fill out the application form on our webpage: `www.sl2college.org/research-collab`
  - mention areas of interests and preferred mentors.
- .. and email your CV to `research-collab@sl2college.org`.
- If we decide to proceed, we ask you to submit a ~ 1 page research statement,
  - your research interests & future plans
  - why you are interested in working with aforesaid mentor.

Application Process

- Fill out the application form on our webpage: www.sl2college.org/research-collab
  - mention areas of interests and preferred mentors.
- .. and email your CV to research-collab@sl2college.org.
- If we decide to proceed, we ask you to submit a ~ 1 page research statement,
  - your research interests & future plans
  - why you are interested in working with aforesaid mentor.
- We send your CV & statement to the mentor. If he/she is interested, we initiate a collaboration.
- You report to us once every 3 months.

SL2College Research Collaboration Team

Ashwin

Nuwan

Rajitha

Umashanthi

Kirthevasan

www.sl2college.org/research-collab

research-collab@sl2college.org

Outline

- Part I: Stochastic bandits (cont'd)
  1. Gaussian processes for smooth bandits
  2. Algorithms: Upper Confidence Bound (UCB) & Thompson Sampling (TS)
- Digression: SL2College Research Collaboration Program
- Part II: My research
  1. Multi-fidelity bandit: cheap approximations to an expensive experiments
  2. Parallelising arm pulls

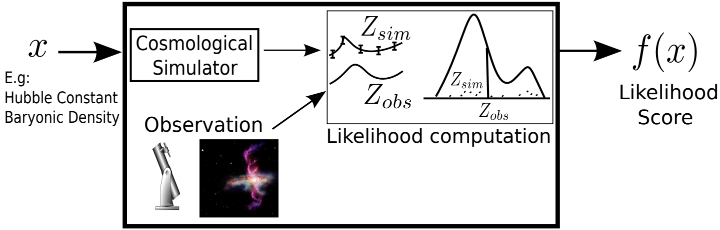
Part 2.1: Multi-fidelity Bandits

Motivating question:
What if we have cheap approximations to f?

Part 2.1: Multi-fidelity Bandits

Motivating question:
What if we have cheap approximations to f?

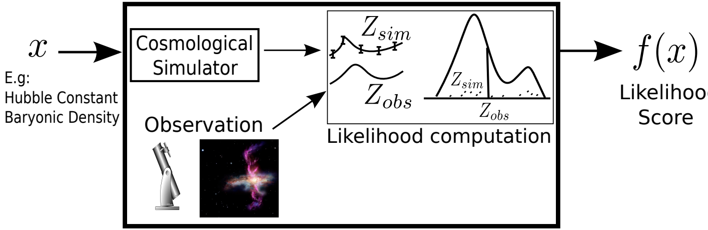
1. Computational astrophysics and other scientific experiments: simulations and numerical computations with less granularity.



Part 2.1: Multi-fidelity Bandits

Motivating question:
What if we have cheap approximations to f?

1. Computational astrophysics and other scientific experiments: simulations and numerical computations with less granularity.



2. Hyper-parameter tuning: Train & validate with a subset of the data.

3. Robotics & autonomous driving: computer simulation vs real world experiment.

Multi-fidelity Methods

For specific applications,

- Industrial design (Forrester et al. 2007)
- Hyper-parameter tuning (Agarwal et al. 2011, Klein et al. 2015, Li et al. 2016)
- Active learning (Zhang & Chaudhuri 2015)
- Robotics (Cutler et al. 2014)

Multi-fidelity bandits & optimisation (Huang et al. 2006, Forrester et al. 2007, March & Wilcox 2012, Poloczek et al. 2016)

Multi-fidelity Methods

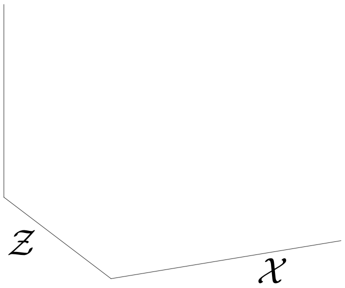
For specific applications,

- Industrial design (Forrester et al. 2007)
- Hyper-parameter tuning (Agarwal et al. 2011, Klein et al. 2015, Li et al. 2016)
- Active learning (Zhang & Chaudhuri 2015)
- Robotics (Cutler et al. 2014)

Multi-fidelity bandits & optimisation (Huang et al. 2006, Forrester et al. 2007, March & Wilcox 2012, Poloczek et al. 2016)

. . . with theoretical guarantees (Kandasamy et al. NIPS 2016a&b, Kandasamy et al. ICML 2017)

Multi-fidelity Bandits (Kandasamy et al. ICML 2017)

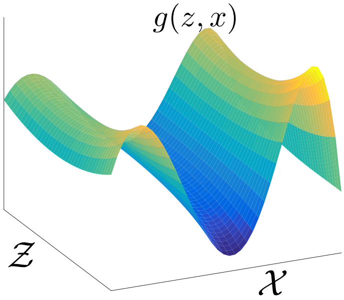


A fidelity space $\mathcal{Z}$ and domain $\mathcal{X}$

$\mathcal{Z} \leftarrow$ all granularity values

$\mathcal{X} \leftarrow$ space of cosmological parameters

Multi-fidelity Bandits (Kandasamy et al. ICML 2017)



A fidelity space $\mathcal{Z}$ and domain $\mathcal{X}$

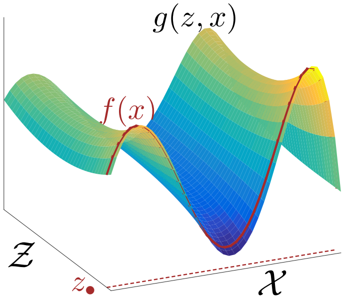
$\mathcal{Z} \leftarrow$ all granularity values

$\mathcal{X} \leftarrow$ space of cosmological parameters

$g : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$.

$g(z,x) \leftarrow$ likelihood score when performing integrations on a grid of size z at cosmological parameters x.

Multi-fidelity Bandits (Kandasamy et al. ICML 2017)



A fidelity space $\mathcal{Z}$ and domain $\mathcal{X}$

$\mathcal{Z} \leftarrow$ all granularity values

$\mathcal{X} \leftarrow$ space of cosmological parameters

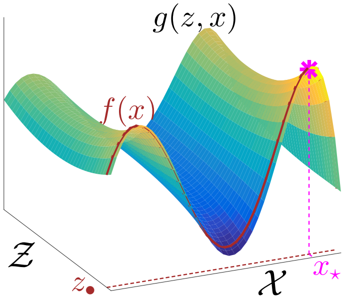
$g : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$.

$g(z,x) \leftarrow$ likelihood score when performing integrations on a grid of size z at cosmological parameters x.

Denote $f(x) = g(z_\bullet, x)$ where $z_\bullet \in \mathcal{Z}$. $z_\bullet$ = highest grid size.

Multi-fidelity Bandits

(Kandasamy et al. ICML 2017)



A fidelity space $\mathcal{Z}$ and domain $\mathcal{X}$

$\mathcal{Z} \leftarrow$ all granularity values

$\mathcal{X} \leftarrow$ space of cosmological parameters

$g : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}$.

$g(z, x) \leftarrow$ likelihood score when performing integrations on a grid of size z at cosmological parameters x.

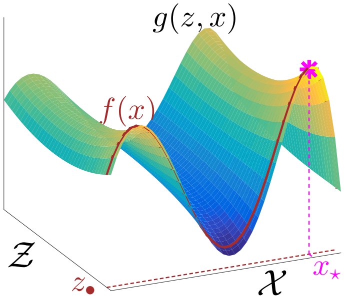
Denote $f(x) = g(z_\bullet, x)$ where $z_\bullet \in \mathcal{Z}$.

$z_\bullet$ = highest grid size.

End Goal: Find $x_\star = \mathrm{argmax}_x\, f(x)$.

Multi-fidelity Bandits

(Kandasamy et al. ICML 2017)



A fidelity space $\mathcal{Z}$ and domain $\mathcal{X}$

$\mathcal{Z} \leftarrow$ all granularity values

$\mathcal{X} \leftarrow$ space of cosmological parameters

$g : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$.

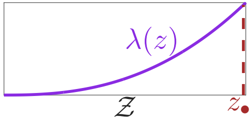
$g(z, x) \leftarrow$ likelihood score when performing integrations on a grid of size z at cosmological parameters x.

Denote $f(x) = g(z_\bullet, x)$ where $z_\bullet \in \mathcal{Z}$. $z_\bullet =$ highest grid size.

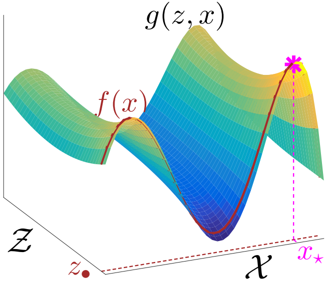
End Goal: Find $x_\star = \mathrm{argmax}_x f(x)$.

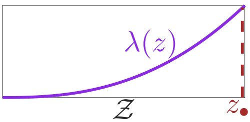
A cost function, $\lambda : \mathcal{Z} \to \mathbb{R}_+$.

$\lambda(z) = \mathcal{O}(z^p)$ (say).



Multi-fidelity Simple Regret (Kandasamy et al. ICML 2017)

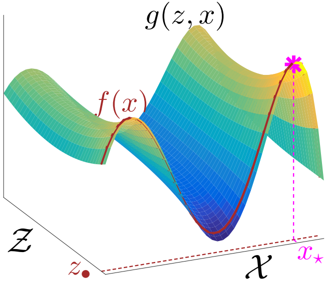


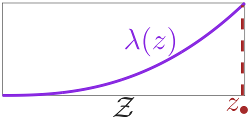


End Goal: Find $x_\star = \mathrm{argmax}_x f(x)$.

Multi-fidelity Simple Regret

(Kandasamy et al. ICML 2017)





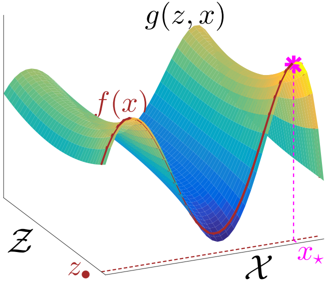
End Goal: Find $x_\star = \mathrm{argmax}_x f(x)$.

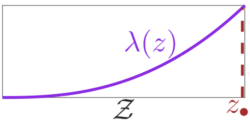
Simple Regret after *capital* Λ: $\quad S(\Lambda) = f(x_\star) - \max_{t:\, z_t = z_\bullet} f(x_t).$

$\Lambda \leftarrow$ amount of a resource spent, e.g. computation time or money.

Multi-fidelity Simple Regret

(Kandasamy et al. ICML 2017)





End Goal: Find $x_\star = \mathrm{argmax}_x f(x)$.

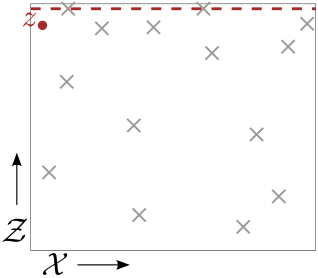
Simple Regret after *capital* Λ: $\quad S(\Lambda) = f(x_\star) - \max_{t:\, z_t = z_\bullet} f(x_t)$.

$\Lambda \leftarrow$ amount of a resource spent, e.g. computation time or money.

No reward for pulling an arm at low fidelities, but use cheap evaluations at $z \neq z_\bullet$ to speed up search for $x_\star$.

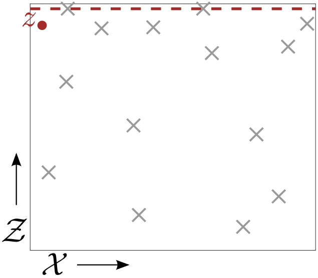
Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Algorithm: BOCA

(Kandasamy et al. ICML 2017)

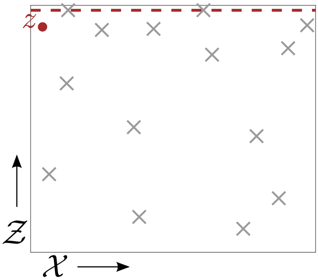


Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

Algorithm: BOCA (Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

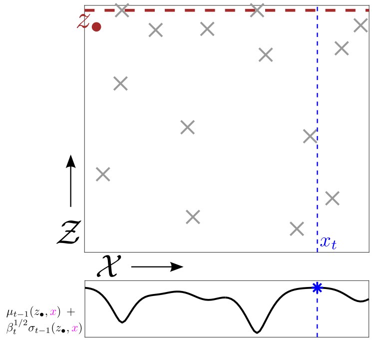
std-dev $\sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2} \sigma_{t-1}(z_\bullet, x)$$

Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

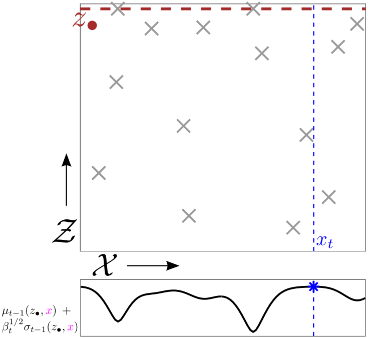
std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

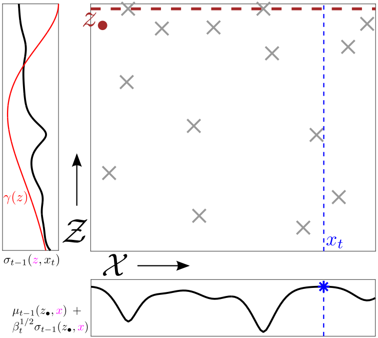
$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

(2) $$\mathcal{Z}_t \approx \{z_\bullet\} \cup \left\{ z \,:\, \sigma_{t-1}(z, x_t) \geq \gamma(z) \right\}$$

(3) $$z_t = \operatorname*{argmin}_{z \in \mathcal{Z}_t} \lambda(z) \qquad \text{(cheapest } z \text{ in } \mathcal{Z}_t\text{)}$$

Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

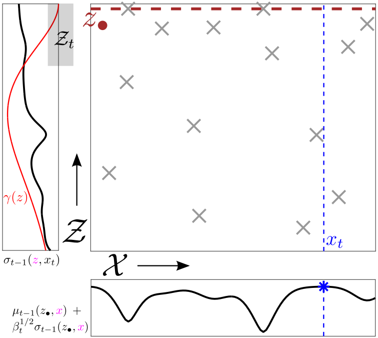
$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

(2) $$\mathcal{Z}_t \approx \{z_\bullet\} \cup \Big\{ z : \sigma_{t-1}(z, x_t) \geq \gamma(z) \Big\}$$

(3) $$z_t = \operatorname*{argmin}_{z \in \mathcal{Z}_t} \lambda(z) \qquad \text{(cheapest } z \text{ in } \mathcal{Z}_t\text{)}$$

Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

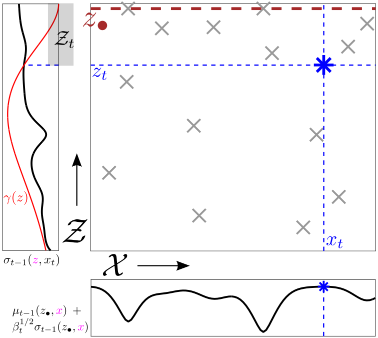
(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

(2) $$\mathcal{Z}_t \approx \{z_\bullet\} \cup \left\{ z : \sigma_{t-1}(z, x_t) \geq \gamma(z) \right\}$$

(3) $$z_t = \operatorname*{argmin}_{z \in \mathcal{Z}_t} \lambda(z) \qquad \text{(cheapest } z \text{ in } \mathcal{Z}_t)$$

Algorithm: BOCA (Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

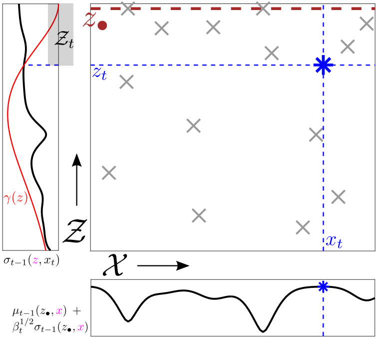
$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

(2) $$\mathcal{Z}_t \approx \{z_\bullet\} \cup \left\{ z : \sigma_{t-1}(z, x_t) \geq \gamma(z) \right\}$$

(3) $$z_t = \operatorname*{argmin}_{z \in \mathcal{Z}_t} \lambda(z) \qquad \text{(cheapest } z \text{ in } \mathcal{Z}_t)$$

Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model $g \sim \mathcal{GP}(0, \kappa)$ and compute posterior $\mathcal{GP}$:

mean $\quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$

std-dev $\quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$

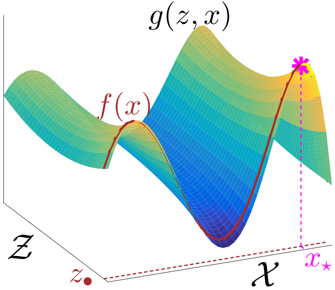
(1) $x_t \leftarrow$ maximise upper confidence bound for $f(x) = g(z_\bullet, x)$.

$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}} \; \mu_{t-1}(z_\bullet, x) + \beta_t^{1/2}\sigma_{t-1}(z_\bullet, x)$$

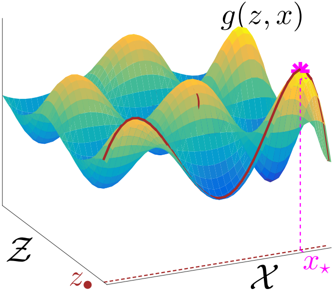
(2) $$\mathcal{Z}_t \approx \{z_\bullet\} \cup \left\{ z : \sigma_{t-1}(z, x_t) \geq \gamma(z) = \left(\frac{\lambda(z)}{\lambda(z_\bullet)}\right)^q \xi(z) \right\}$$

(3) $$z_t = \operatorname*{argmin}_{z \in \mathcal{Z}_t} \lambda(z) \qquad \text{(cheapest } z \text{ in } \mathcal{Z}_t\text{)}$$

Theoretical Results for BOCA

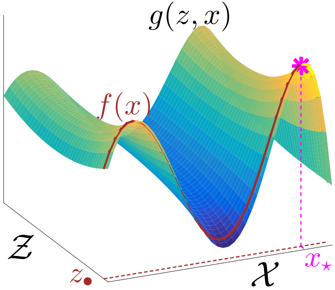


“good”

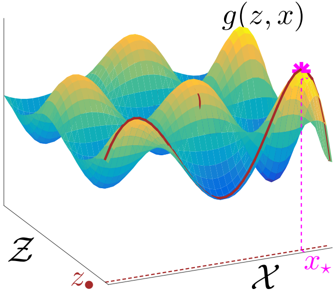


“bad”

Theoretical Results for BOCA



"good"
large $h_{\mathcal{Z}}$



"bad"
small $h_{\mathcal{Z}}$

E.g.: For SE kernels, bandwidth $h_{\mathcal{Z}}$ controls smoothness.

Theoretical Results for BOCA

GP-UCB SE kernel, (Srinivas et al. 2010)

$$\text{w.h.p} \qquad S(\Lambda) \;\lesssim\; \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda}}$$

Theoretical Results for BOCA

GP-UCB SE kernel, (Srinivas et al. 2010)

$$\text{w.h.p} \qquad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda}}$$

BOCA SE kernel, (Kandasamy et al. ICML 2017)

$$\text{w.h.p} \quad \forall \alpha > 0, \quad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}_\alpha)}{\Lambda}} + \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda^{2-\alpha}}}$$

$$\mathcal{X}_\alpha = \left\{ x;\ f(x_\star) - f(x) \lesssim C_\alpha \frac{1}{h_{\mathcal{Z}}} \right\}$$

Theoretical Results for BOCA

GP-UCB   SE kernel,   (Srinivas et al. 2010)

$$\text{w.h.p} \qquad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda}}$$

BOCA   SE kernel,   (Kandasamy et al. ICML 2017)

$$\text{w.h.p} \quad \forall \alpha > 0, \quad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}_\alpha)}{\Lambda}} + \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda^{2-\alpha}}}$$

$$\mathcal{X}_\alpha = \left\{ x;\ f(x_\star) - f(x) \lesssim C_\alpha \frac{1}{h_\mathcal{Z}} \right\}$$

If $h_\mathcal{Z}$ is large (good approximations), $\mathrm{vol}(\mathcal{X}_\alpha) \ll \mathrm{vol}(\mathcal{X})$, and BOCA is much better than GP-UCB.

Theoretical Results for BOCA

GP-UCB   SE kernel,   (Srinivas et al. 2010)

$$\text{w.h.p} \qquad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda}}$$

BOCA   SE kernel,   (Kandasamy et al. ICML 2017)

$$\text{w.h.p} \quad \forall \alpha > 0, \quad S(\Lambda) \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}_\alpha)}{\Lambda}} + \sqrt{\frac{\mathrm{vol}(\mathcal{X})}{\Lambda^{2-\alpha}}}$$

$$\mathcal{X}_\alpha = \left\{ x;\ f(x_\star) - f(x) \lesssim C_\alpha \frac{1}{h_\mathcal{Z}} \right\}$$

If $h_\mathcal{Z}$ is large (good approximations), $\mathrm{vol}(\mathcal{X}_\alpha) \ll \mathrm{vol}(\mathcal{X})$, and BOCA is much better than GP-UCB.

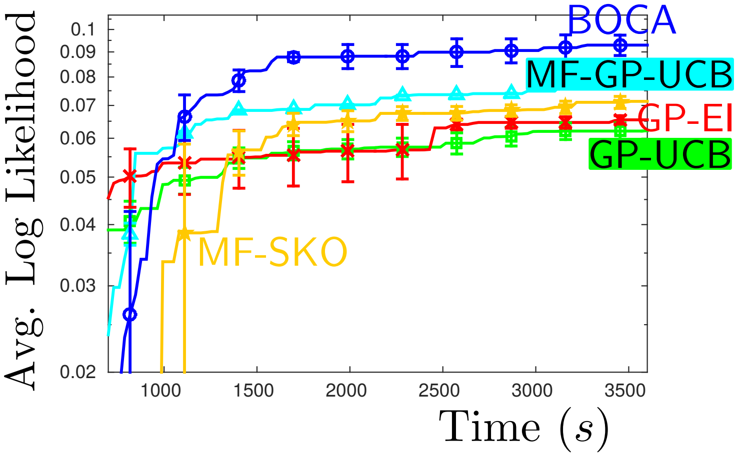
N.B: Dropping constants and $\mathrm{polylog}$ terms.

Experiment: Cosmological inference on Type-1a supernovae data

Estimate Hubble constant, dark matter fraction & dark energy fraction by maximising likelihood on $N_\bullet = 192$ data.
Requires numerical integration on a grid of size $G_\bullet = 10^6$.
Approximate with $N \in [50, 192]$ or $G \in [10^2, 10^6]$ (2D fidelity space).

Experiment: Cosmological inference on Type-1a supernovae data

Estimate Hubble constant, dark matter fraction & dark energy fraction by maximising likelihood on $N_\bullet = 192$ data.
Requires numerical integration on a grid of size $G_\bullet = 10^6$.
Approximate with $N \in [50, 192]$ or $G \in [10^2, 10^6]$ (2D fidelity space).

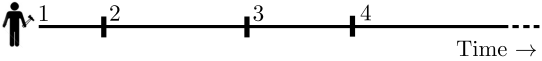


Outline

- Part I: Stochastic bandits (cont'd)
  1. Gaussian processes for smooth bandits
  2. Algorithms: Upper Confidence Bound (UCB) & Thompson Sampling (TS)
- Digression: SL2College Research Collaboration Program
- Part II: My research
  1. Multi-fidelity bandit: cheap approximations to an expensive experiments
  2. Parallelising arm pulls

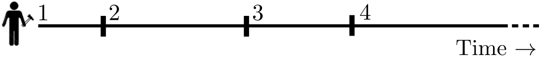
Part 2.2: Parallelising arm pulls

Sequential arm pulls with one worker

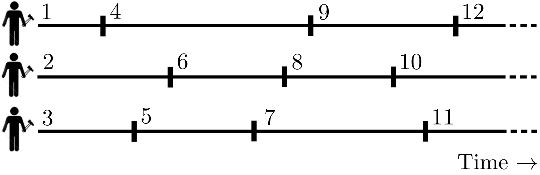


Part 2.2: Parallelising arm pulls

Sequential arm pulls with one worker

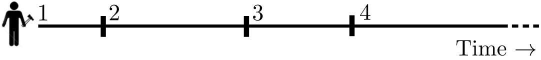


Parallel arm pulls with M workers (Asynchronous)

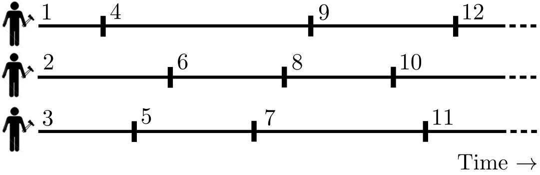


Part 2.2: Parallelising arm pulls

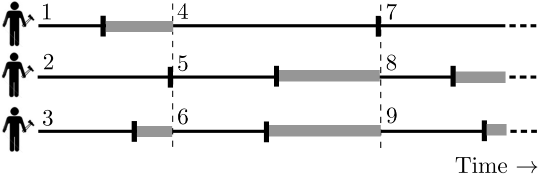
Sequential arm pulls with one worker



Parallel arm pulls with *M* workers (Asynchronous)



Parallel arm pulls with *M* workers (Synchronous)



Why parallelisation?

- **Computational experiments:** infrastructure with 100-1000's CPUs or GPUs.
- **Drug discovery:** High throughput screening

Why parallelisation?

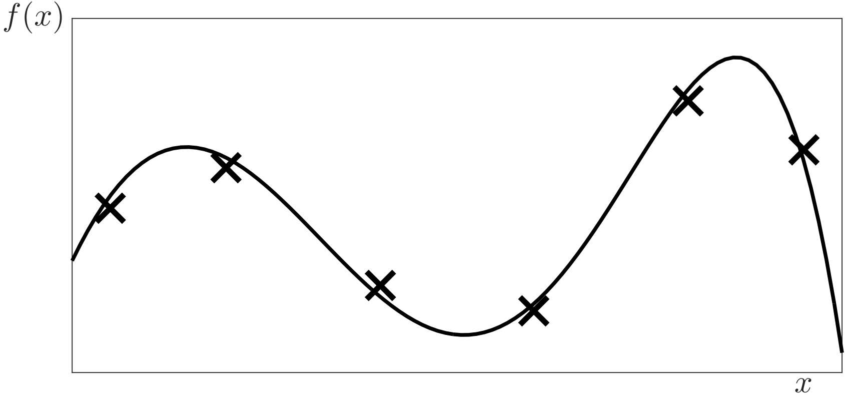
- **Computational experiments:** infrastructure with 100-1000's CPUs or GPUs.
- **Drug discovery:** High throughput screening

Prior work: (Ginsbourger et al. 2011, Janusevskis et al. 2012, Wang et al. 2016, González et al. 2015, Desautels et al. 2014, Contal et al. 2013, Shah and Ghahramani 2015, Kathuria et al. 2016, Wang et al. 2017, Wu and Frazier 2016, Hernandez-Lobato et al. 2017)

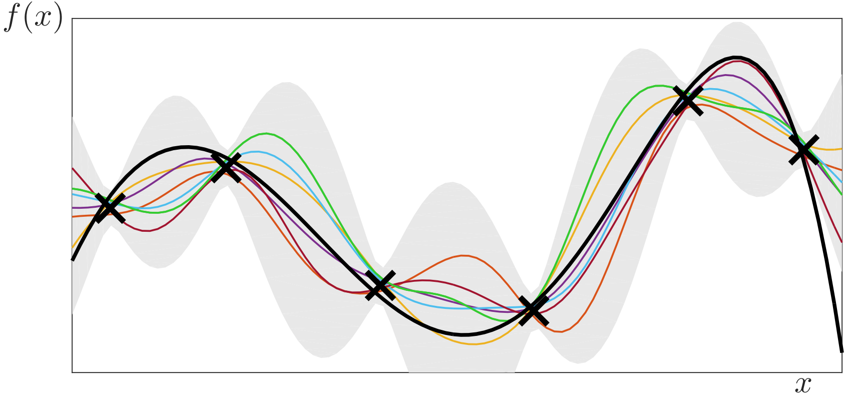
Shortcomings

- **Asynchronicity**
- **Theoretical guarantees**
- **Computationally & conceptually simple**

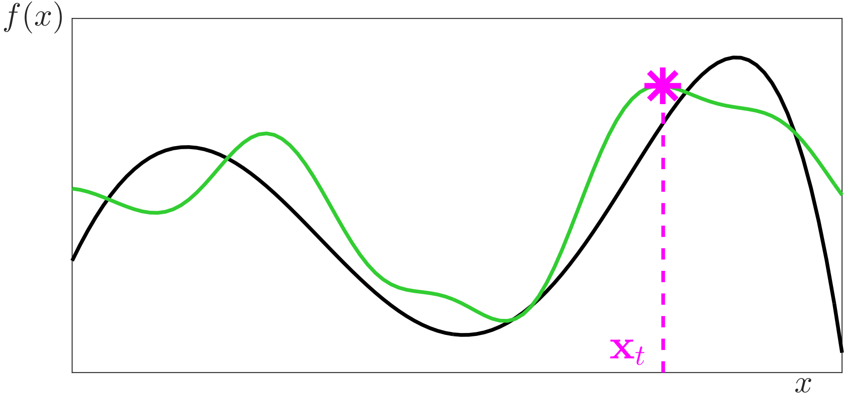
Review: Sequential Thompson Sampling in GP Bandits



Review: Sequential Thompson Sampling in GP Bandits



Review: Sequential Thompson Sampling in GP Bandits



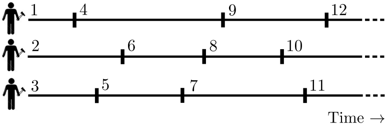
Draw sample g from posterior. Choose $x_t = \mathrm{argmax}_x\, g(x)$.

Parallelised Thompson Sampling (Kandasamy et al. Arxiv 2017)

Asynchronous: asyTS

At any given time,

1. $(x', y') \leftarrow$ Wait for a worker to finish.
2. Compute posterior $\mathcal{GP}$.
3. Draw a sample $g \sim \mathcal{GP}$.
4. Re-deploy worker at $\mathrm{argmax}\, g$.

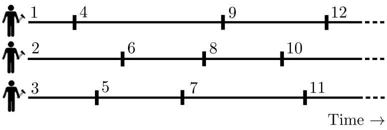


Parallelised Thompson Sampling (Kandasamy et al. Arxiv 2017)

Asynchronous: asyTS

At any given time,

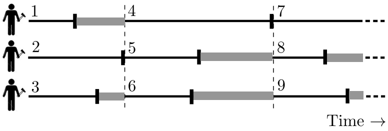
1. (x', y') ← Wait for a worker to finish.
2. Compute posterior $\mathcal{GP}$.
3. Draw a sample $g \sim \mathcal{GP}$.
4. Re-deploy worker at $\operatorname{argmax} g$.



Synchronous: synTS

At any given time,

1. $\{(x'_m, y'_m)\}_{m=1}^{M}$ ← Wait for all workers to finish.
2. Compute posterior $\mathcal{GP}$.
3. Draw M samples $g_m \sim \mathcal{GP}$, $\forall m$.
4. Re-deploy worker m at $\operatorname{argmax} g_m$, $\forall m$.



Theoretical Results: number of evaluations

Sequential TS, SE Kernel (Russo & van Roy 2014)

$$\mathbb{E}[S_n] \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

Theoretical Results: number of evaluations

Sequential TS, SE Kernel (Russo & van Roy 2014)

$$\mathbb{E}[S_n] \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

Theorem: synTS & asyTS, SE Kernel (Kandasamy et al. Arxiv 2017)

$$\mathbb{E}[S_n] \lesssim \frac{M \log(M)^{2d}}{n} + \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

$n \leftarrow \#$ completed arm pulls by all workers.

Theoretical Results: number of evaluations

Sequential TS, SE Kernel (Russo & van Roy 2014)

$$\mathbb{E}[S_n] \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

Theorem: synTS & asyTS, SE Kernel (Kandasamy et al. Arxiv 2017)

$$\mathbb{E}[S_n] \lesssim \frac{M \log(M)^{2d}}{n} + \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

$n \leftarrow \#$ completed arm pulls by all workers.

Why is this interesting?

- A sequential algorithm can make use of information from all previous rounds to determine where to evaluate next.
- A parallel algorithm could be missing up to $M - 1$ results at any given time.

Theoretical Results: number of evaluations

Sequential TS, SE Kernel (Russo & van Roy 2014)

$$\mathbb{E}[S_n] \lesssim \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

Theorem: synTS & asyTS, SE Kernel (Kandasamy et al. Arxiv 2017)

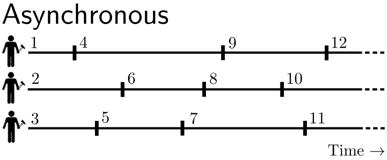
$$\mathbb{E}[S_n] \lesssim \frac{M \log(M)^{2d}}{n} + \sqrt{\frac{\mathrm{vol}(\mathcal{X}) \log(n)^d}{n}}$$

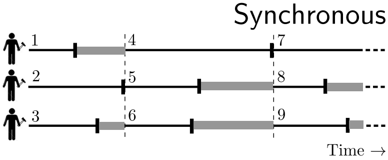
$n \leftarrow \#$ completed arm pulls by all workers.

Why is this interesting?

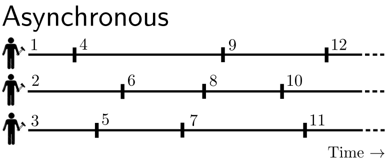
- A sequential algorithm can make use of information from all previous rounds to determine where to evaluate next.
- A parallel algorithm could be missing up to $M - 1$ results at any given time. But randomisation helps!

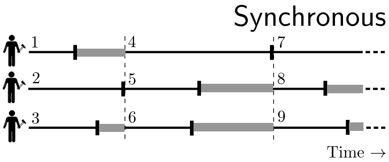
Theoretical Results: Simple regret with time





Theoretical Results: Simple regret with time



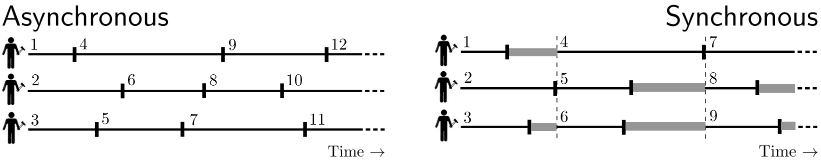


Theorem (Informal) (Kandasamy et al. Arxiv 2017)

If evaluation times are the same, asyTS ≈ synTS.

Otherwise, bounds for asyTS is strictly better than synTS. More the variability in evaluation times, the bigger the difference.

Theoretical Results: Simple regret with time



Theorem (Informal) (Kandasamy et al. Arxiv 2017)

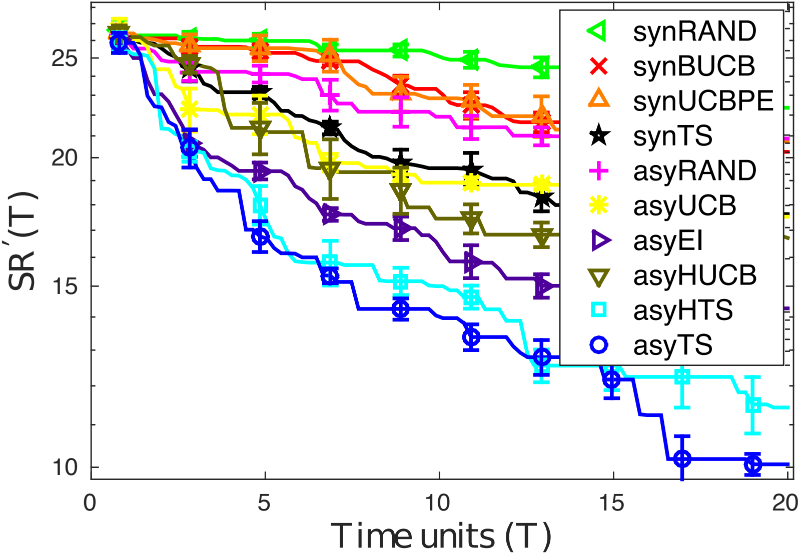
If evaluation times are the same, asyTS ≈ synTS.

Otherwise, bounds for asyTS is strictly better than synTS. More the variability in evaluation times, the bigger the difference.

- Bounded tail decay: constant factor
- Sub-gaussian tail decay: $\sqrt{\log(M)}$ factor
- Sub-exponential tail decay: $\log(M)$ factor

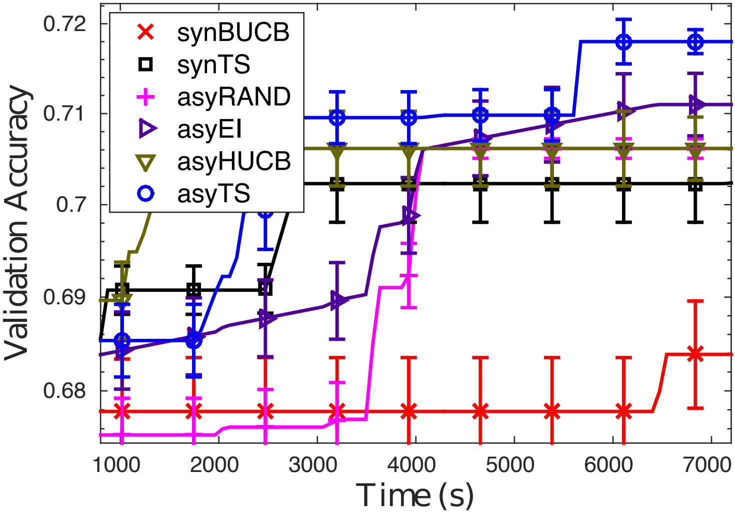
Experiment: Currin-Exponential-14D $M = 35$

Evaluation time sampled from a Pareto-3 distribution



Experiment: Hyper-parameter tuning in Cifar10 $M = 4$

Tune $\#$ filters in in range $(32, 256)$ for each layer in a 6 layer CNN.
Time taken for an evaluation: 4 - 16 minutes.



Summary

- Bandits are a framework for studying exploration vs exploitation trade-offs when optimising black-box functions.
- Smooth bandit formulations are more common in practical applications.
- Several algorithms: UCB, TS, Index based policies, ϵ-greedy etc.

Summary

- Bandits are a framework for studying exploration vs exploitation trade-offs when optimising black-box functions.
- Smooth bandit formulations are more common in practical applications.
- Several algorithms: UCB, TS, Index based policies, ϵ-greedy etc.
- **Multi-fidelity Bandits:** Allows us to use cheap approximations to a an expensive experiment to quickly find the optimum.
- **Parallelised TS:** Simple and intuitive way to deal with multiple workers.

Akshay

Barnabás

Gautam

Jeff

Junier

Thank You

Slides: `www.cs.cmu.edu/~kkandasa/misc/mora-slides.pdf`