High Dimensional Bayesian Optimisation and Bandits via Additive Models

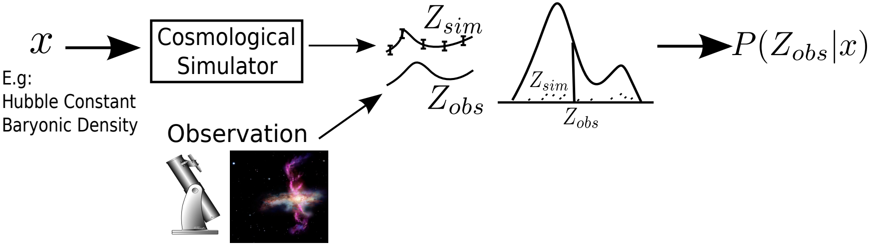
Kirthevasan Kanda**samy**, Jeff Schneider, Barnabás Póczos

ICML '15

July 8 2015

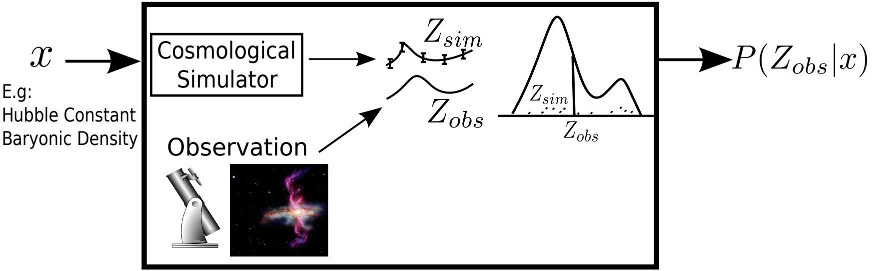
Bandits & Optimisation

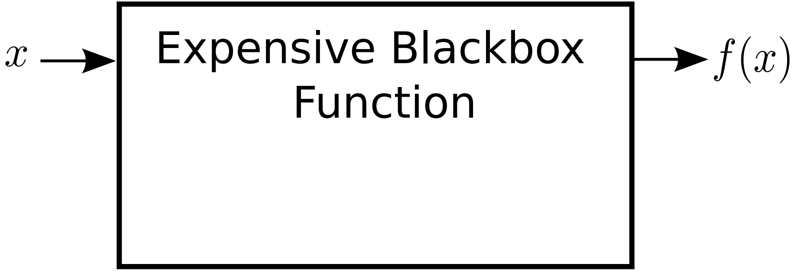
Maximum Likelihood inference in Computational Astrophysics



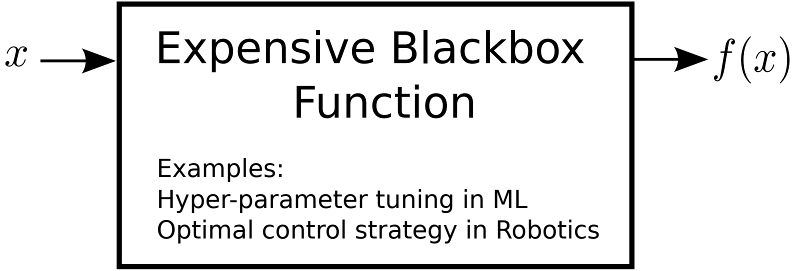
Bandits & Optimisation

Maximum Likelihood inference in Computational Astrophysics



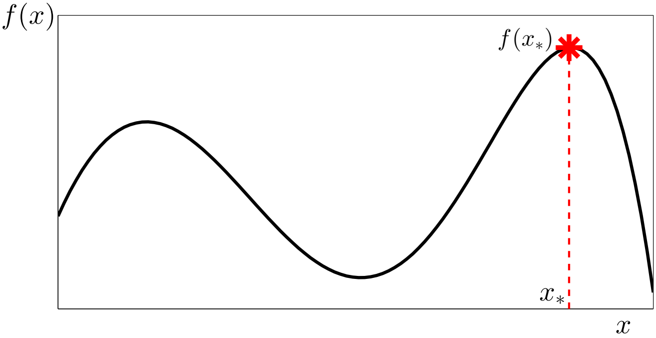

x
Expensive Blackbox Function
$f(x)$

Bandits & Optimisation



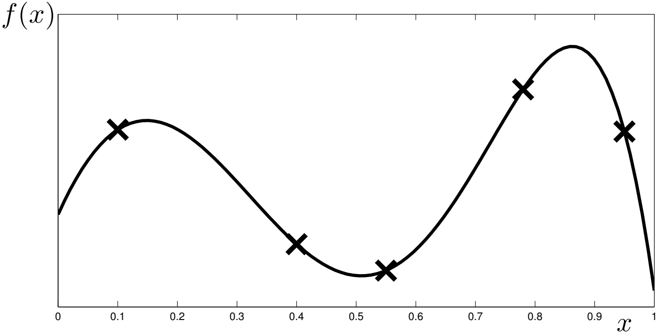
Bandits & Optimisation

$f : [0,1]^D \to \mathbb{R}$ is an expensive, black-box, nonconvex function.
Let $\mathbf{x}_* = \operatorname{argmax}_x f(x)$.



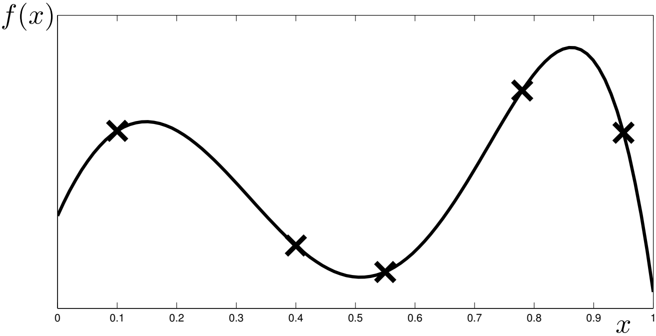
Bandits & Optimisation

$f : [0, 1]^D \to \mathbb{R}$ is an expensive, black-box, nonconvex function.
Let $\mathbf{x}_* = \mathrm{argmax}_x\, f(x)$.



Bandits & Optimisation

$f : [0,1]^D \to \mathbb{R}$ is an expensive, black-box, nonconvex function.
Let $\mathbf{x}_* = \mathrm{argmax}_x f(x)$.

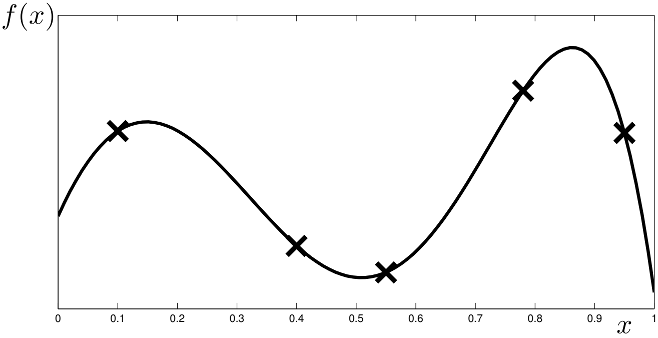


Optimisation $\cong$ Minimise *Simple Regret*.

$$S_T = f(\mathbf{x}_*) - \max_{\mathbf{x}_t,\, t=1,\dots,T} f(\mathbf{x}_t).$$

Bandits & Optimisation

$f : [0,1]^D \to \mathbb{R}$ is an expensive, black-box, nonconvex function.
Let $\mathbf{x}_* = \mathrm{argmax}_x\, f(x)$.

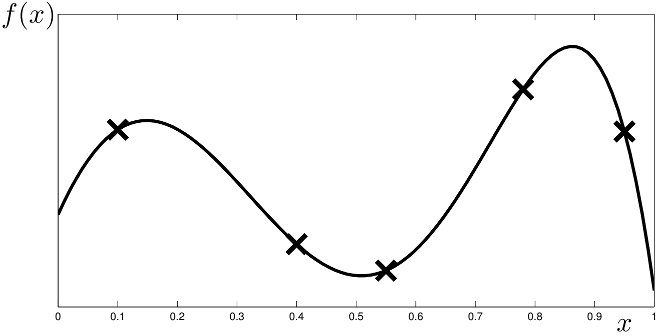


Bandits $\cong$ Minimise *Cumulative Regret*.

$$R_T = \sum_{t=1}^{T} f(\mathbf{x}_*) - f(\mathbf{x}_t).$$

Bandits & Optimisation

$f : [0,1]^D \to \mathbb{R}$ is an expensive, black-box, nonconvex function.
Let $\mathbf{x}_* = \operatorname{argmax}_x f(x)$.

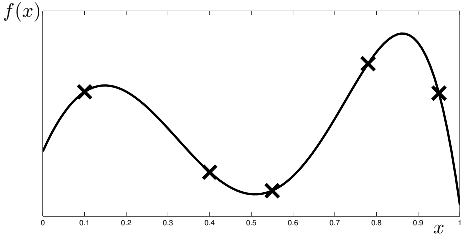


Optimisation $\cong$ Minimise *Simple Regret*.

$$S_T = f(\mathbf{x}_*) - \max_{\mathbf{x}_t,\, t=1,\ldots,T} f(\mathbf{x}_t).$$

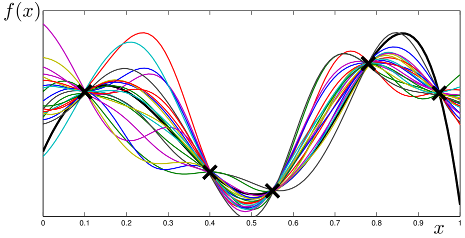
Gaussian Process (Bayesian) Optimisation

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



Gaussian Process (Bayesian) Optimisation

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

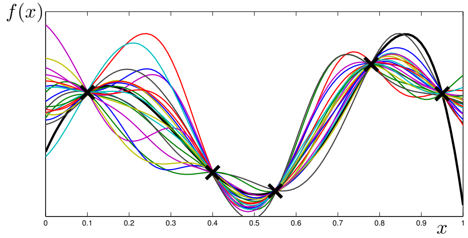


Obtain posterior GP.

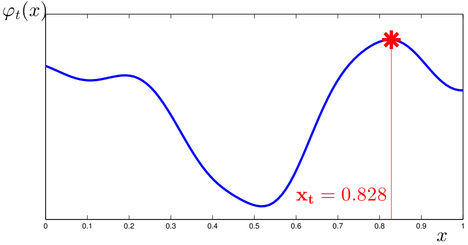
.

Gaussian Process (Bayesian) Optimisation

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



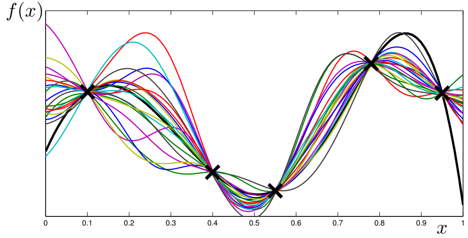
Maximise acquisition function φ_t: $\mathbf{x}_t = \mathrm{argmax}_x \, \varphi_t(x)$.



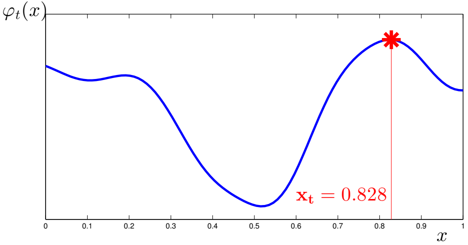
GP-UCB: $\varphi_t(x) = \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x)$ (Srinivas et al. 2010)

Gaussian Process (Bayesian) Optimisation

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



Maximise acquisition function φ_t: $\mathbf{x}_t = \mathrm{argmax}_x \, \varphi_t(x)$.



φ_t: Expected Improvement (**GP-EI**), Thompson Sampling etc.

Scaling to Higher Dimensions

Two Key Challenges:

- **Statistical Difficulty:**
  Nonparametric sample complexity exponential in D.
- **Computational Difficulty:**
  Optimising φ_t to within ζ accuracy requires $\mathcal{O}(\zeta^{-D})$ effort.

Scaling to Higher Dimensions

Two Key Challenges:

- **Statistical Difficulty:**
  Nonparametric sample complexity exponential in D.
- **Computational Difficulty:**
  Optimising φ_t to within ζ accuracy requires $\mathcal{O}(\zeta^{-D})$ effort.

Existing Work:

- (Chen et al. 2012): f depends on a small number of variables. Find variables and then **GP-UCB**.
- (Wang et al. 2013): f varies along a lower dimensional subspace. **GP-EI** on a random subspace.
- (Djolonga et al. 2013): f varies along a lower dimensional subspace. Find subspace and then **GP-UCB**.

Scaling to Higher Dimensions

Two Key Challenges:

- **Statistical Difficulty:**
  Nonparametric sample complexity exponential in D.

- **Computational Difficulty:**
  Optimising φ_t to within ζ accuracy requires $\mathcal{O}(\zeta^{-D})$ effort.

Existing Work:
Chen et al. 2012, Wang et al. 2013, Djolonga et al. 2013.

- Assumes f varies only along a low dimensional subspace.
- Perform BO on a low dimensional subspace.
- Assumption too strong in realistic settings.

Additive Functions

Structural assumption:

$$f(x) = f^{(1)}(x^{(1)}) + f^{(2)}(x^{(2)}) + \ldots + f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0,1]^d, \qquad d \ll D, \qquad x^{(i)} \cap x^{(j)} = \varnothing.$$

Additive Functions

Structural assumption:

$$f(x) \;=\; f^{(1)}(x^{(1)}) \;+\; f^{(2)}(x^{(2)}) \;+\; \ldots \;+\; f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0,1]^d, \qquad d \ll D, \qquad x^{(i)} \cap x^{(j)} = \varnothing.$$

E.g. $f(x_{\{1,\ldots,10\}}) = \boxed{f^{(1)}(x_{\{1,3,9\}})} + \boxed{f^{(2)}(x_{\{2,4,8\}})} + \boxed{f^{(3)}(x_{\{5,6,10\}})}$.

1	2	3	4	5	6	~~7~~	8	9	10

Call $\{\mathcal{X}^{(j)}\}_{j=1}^{M} = \{(1,3,9),(2,4,8),(5,6,10)\}$ the "decomposition".

Additive Functions

Structural assumption:

$$f(x) = f^{(1)}(x^{(1)}) + f^{(2)}(x^{(2)}) + \ldots + f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0,1]^d, \qquad d \ll D, \qquad x^{(i)} \cap x^{(j)} = \varnothing.$$

Assume each $f^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$. Then $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where,

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x^{(1)'}) + \cdots + \kappa^{(M)}(x^{(M)}, x^{(M)'}).$$

Additive Functions

Structural assumption:

$$f(x) \;=\; f^{(1)}(x^{(1)}) \,+\, f^{(2)}(x^{(2)}) \,+\, \dots \,+\, f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0,1]^d, \qquad d \ll D, \qquad x^{(i)} \cap x^{(j)} = \varnothing.$$

Assume each $f^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$. Then $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where,

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x^{(1)'}) + \cdots + \kappa^{(M)}(x^{(M)}, x^{(M)'}).$$

Given $(X, Y) = \{(x_i, y_i)_{i=1}^T\}$, and test point $x_\dagger$,

$$f^{(j)}(x_\dagger^{(j)})|X, Y \sim \mathcal{N}(\mu^{(j)}, {\sigma^{(j)}}^2).$$

Outline

1. **GP-UCB**
2. The **Add-GP-UCB** algorithm
   - **Bounds on S_T: exponential in $D \rightarrow$ linear in D.**
   - **An easy-to-optimise acquisition function.**
   - **Performs well even when f is not additive.**
3. Experiments
4. Conclusion & some open questions

GP-UCB

$$\mathbf{x}_t = \operatorname*{argmax}_{x \in \mathcal{X}} \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x)$$

GP-UCB

$$\mathbf{x}_t = \operatorname*{argmax}_{x \in \mathcal{X}} \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x)$$

Squared Exponential Kernel

$$\kappa(x, x') = A \exp\left(\frac{\|x - x'\|^2}{2h^2}\right)$$

Theorem (Srinivas et al. 2010)

Let $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$. Then w.h.p,

$$S_T \in \mathcal{O}\left(\sqrt{\frac{D^D (\log T)^D}{T}}\right).$$

GP-UCB on additive κ

If $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x^{(1)'}) + \cdots + \kappa^{(M)}(x^{(M)}, x^{(M)'}).$$

$\kappa^{(j)} \rightarrow$ SE Kernel.

GP-UCB on additive κ

If $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x^{(1)'}) + \cdots + \kappa^{(M)}(x^{(M)}, x^{(M)'}).$$

$\kappa^{(j)} \rightarrow$ SE Kernel.

Can be shown: If each $\kappa^{(j)}$ is a SE kernel,

$$S_T \in \mathcal{O}\left(\sqrt{\frac{D^2 d^d (\log T)^d}{T}}\right).$$

GP-UCB on additive κ

If $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$ where

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x^{(1)'}) + \cdots + \kappa^{(M)}(x^{(M)}, x^{(M)'}).$$

$\kappa^{(j)} \rightarrow$ SE Kernel.

Can be shown: If each $\kappa^{(j)}$ is a SE kernel,

$$S_T \in \mathcal{O}\left(\sqrt{\frac{D^2 d^d (\log T)^d}{T}}\right).$$

But $\varphi_t = \mu_{t-1} + \beta_t^{1/2}\sigma_{t-1}$ is D-dimensional !

Add-GP-UCB

$$\widetilde{\varphi}_t(x) = \sum_{j=1}^{M} \mu_{t-1}^{(j)}(x) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(x^{(j)}).$$

Add-GP-UCB

$$\widetilde{\varphi}_t(x) = \sum_{j=1}^{M} \underbrace{\mu_{t-1}^{(j)}(x) + \beta_t^{1/2}\sigma_{t-1}^{(j)}(x^{(j)})}_{\widetilde{\varphi}_t^{(j)}(x^{(j)})}.$$

Maximise each $\widetilde{\varphi}_t^{(j)}$ separately.

Requires only $\mathcal{O}(\mathrm{poly}(D)\zeta^{-d})$ effort (vs $\mathcal{O}(\zeta^{-D})$ for **GP-UCB**).

Add-GP-UCB

$$\widetilde{\varphi}_t(x) = \sum_{j=1}^{M} \underbrace{\mu_{t-1}^{(j)}(x) + \beta_t^{1/2}\sigma_{t-1}^{(j)}(x^{(j)})}_{\widetilde{\varphi}_t^{(j)}(x^{(j)})}.$$

Maximise each $\widetilde{\varphi}_t^{(j)}$ separately.

Requires only $\mathcal{O}(\mathrm{poly}(D)\zeta^{-d})$ effort (vs $\mathcal{O}(\zeta^{-D})$ for **GP-UCB**).

Theorem

Let $f^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$ and $f = \sum_j f^{(j)}$. Then w.h.p,

$$S_T \in \mathcal{O}\left(\sqrt{\frac{D^2 d^d (\log T)^d}{T}}\right).$$

Summary of Theoretical Results (for SE Kernel)

GP-UCB with no assumption on f:

$$S_T \in \mathcal{O}\left(D^{D/2} (\log T)^{D/2} T^{-1/2} \right)$$

GP-UCB on additive f:

$$S_T \in \mathcal{O}\left(D T^{-1/2} \right)$$

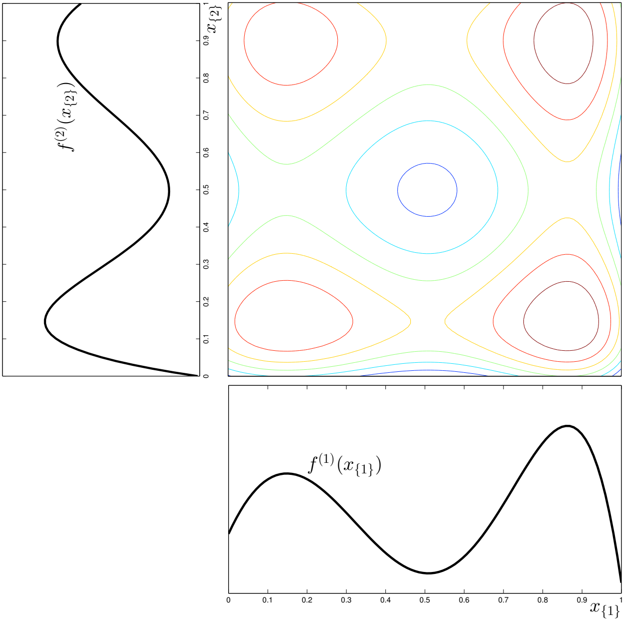
Maximising φ_t : $\mathcal{O}(\zeta^{-D})$ effort.

Add-GP-UCB on additive f:

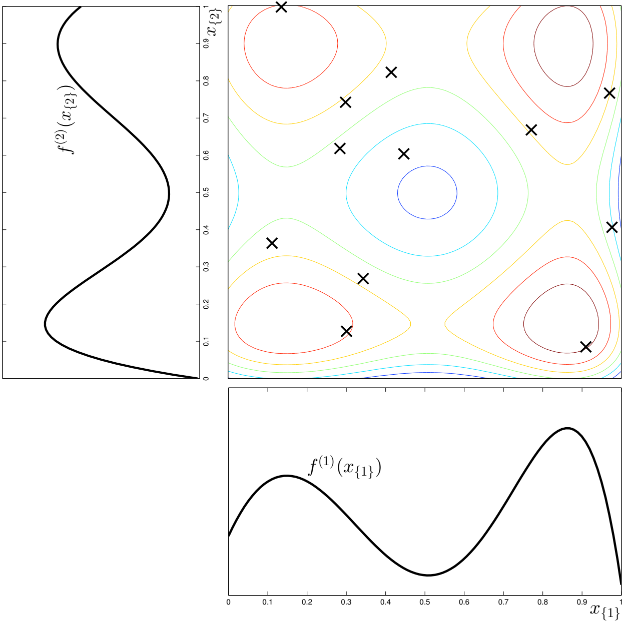
$$S_T \in \mathcal{O}\left(D T^{-1/2} \right)$$

Maximising $\widetilde{\varphi}_t$: $\mathcal{O}(\mathrm{poly}(D)\zeta^{-d})$ effort.

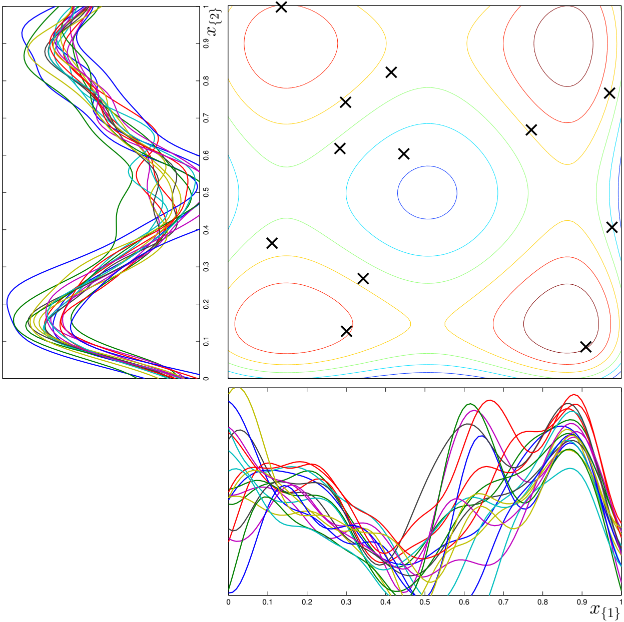
Add-GP-UCB $f(x_{\{1,2\}}) = f^{(1)}(x_{\{1\}}) + f^{(2)}(x_{\{2\}})$



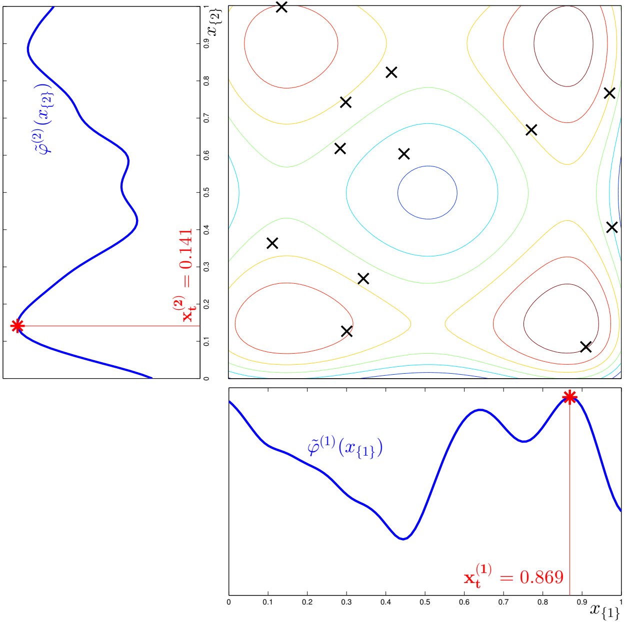
Add-GP-UCB $f(x_{\{1,2\}}) = f^{(1)}(x_{\{1\}}) + f^{(2)}(x_{\{2\}})$



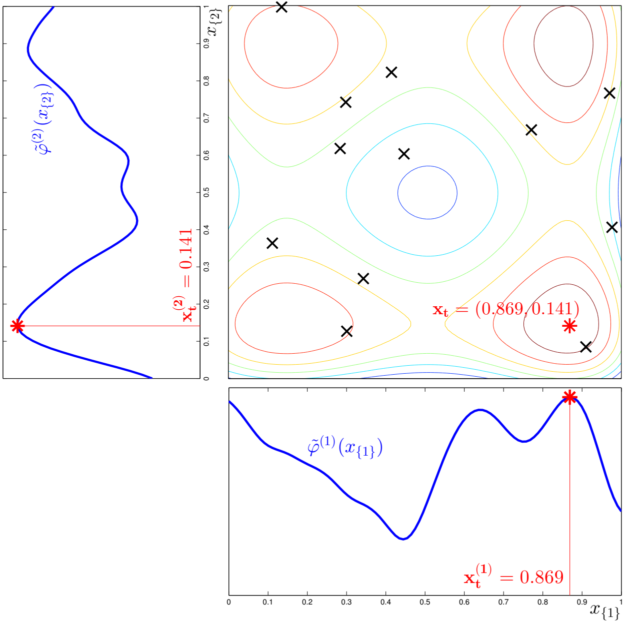
Add-GP-UCB $f(x_{\{1,2\}}) = f^{(1)}(x_{\{1\}}) + f^{(2)}(x_{\{2\}})$



Add-GP-UCB $f(x_{\{1,2\}}) = f^{(1)}(x_{\{1\}}) + f^{(2)}(x_{\{2\}})$



Add-GP-UCB $f(x_{\{1,2\}}) = f^{(1)}(x_{\{1\}}) + f^{(2)}(x_{\{2\}})$



Additive modeling in non-additive settings

- Additive models common in high dimensional regression.
  E.g.: Backfitting, MARS, COSSO, RODEO, SpAM etc.
  $f(x_{\{1,\ldots,D\}}) = f(x_{\{1\}}) + f(x_{\{2\}}) + \cdots + f(x_{\{D\}})$.

Additive modeling in non-additive settings

- Additive models common in high dimensional regression.
  E.g.: Backfitting, MARS, COSSO, RODEO, SpAM etc.
  $f(x_{\{1,\dots,D\}}) = f(x_{\{1\}}) + f(x_{\{2\}}) + \cdots + f(x_{\{D\}})$.
- Additive models are *statistically* simpler $\implies$ worse bias, but much better variance in low sample regime.

Additive modeling in non-additive settings

- Additive models common in high dimensional regression.
  E.g.: Backfitting, MARS, COSSO, RODEO, SpAM etc.
  $f(x_{\{1,\dots,D\}}) = f(x_{\{1\}}) + f(x_{\{2\}}) + \dots + f(x_{\{D\}})$.
- Additive models are *statistically* simpler $\implies$ worse bias, but much better variance in low sample regime.
- In BO applications queries are *expensive*. So we usually cannot afford many queries.

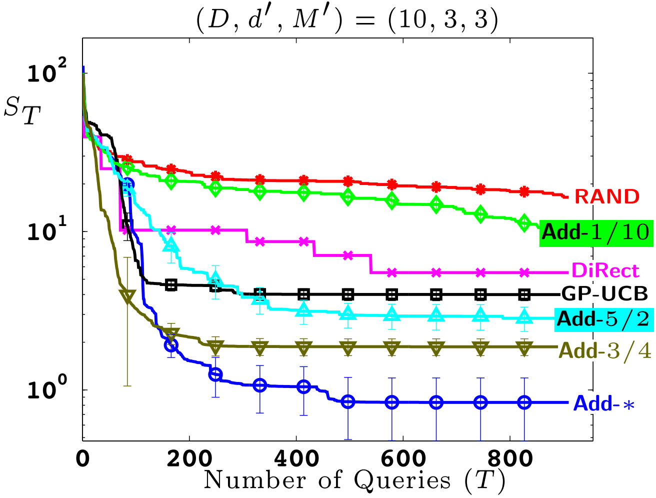
Additive modeling in non-additive settings

- Additive models common in high dimensional regression. E.g.: Backfitting, MARS, COSSO, RODEO, SpAM etc. $f(x_{\{1,\dots,D\}}) = f(x_{\{1\}}) + f(x_{\{2\}}) + \dots + f(x_{\{D\}})$.
- Additive models are *statistically* simpler $\implies$ worse bias, but much better variance in low sample regime.
- In BO applications queries are *expensive*. So we usually cannot afford many queries.
- **Observation:**
  Add-GP-UCB does well even when f is not additive.
  - Better bias/ variance trade-off in high dimensional regression.
  - Easy to maximise acquisition function.

Unknown Kernel/ Decomposition in practice

Learn kernel hyper-parameters and decomposition $\{\mathcal{X}_j\}$ by maximising GP marginal likelihood periodically.

Experiments

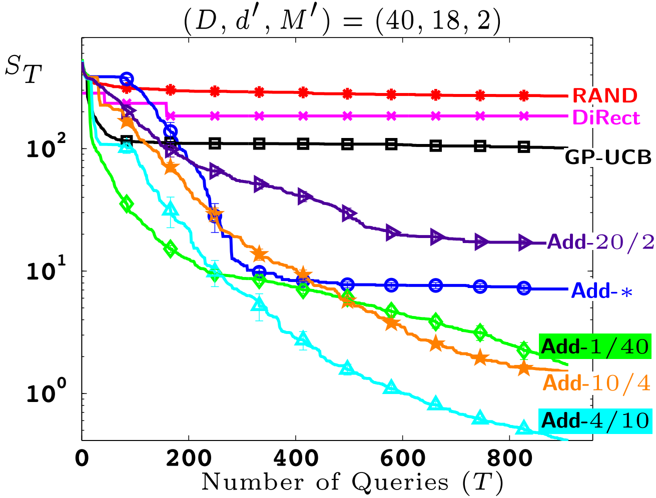


Add-$*$: Knows decomposition.

Add-d/M: M groups of size $\leq d$.

Use **1000** DiRect evaluations to maximise acquisition function.
DiRect: **Di**viding **Rect**angles (Jones et al. 1993)

Experiments

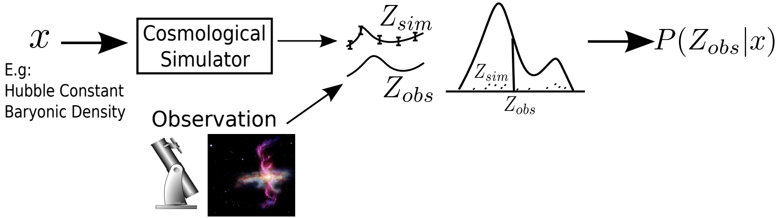


Add-$*$: Knows decomposition.

Add-d/M: M groups of size $\leq d$.

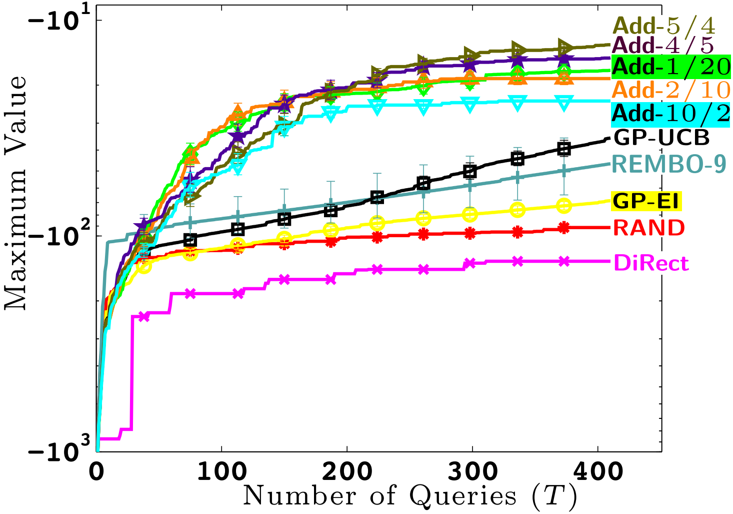
Use **4000** DiRect evaluations to maximise acquisition function.

SDSS Luminous Red Galaxies



- **Task:** Find maximum likelihood cosmological parameters.
- 20 Dimensions. But only 9 parameters are relevant.
- Each query takes 2-5 seconds.
- Use 500 DiRect evaluations to maximise acquisition function.

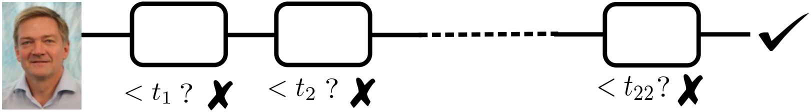
SDSS Luminous Red Galaxies



REMBO: (Wang et al. 2013)

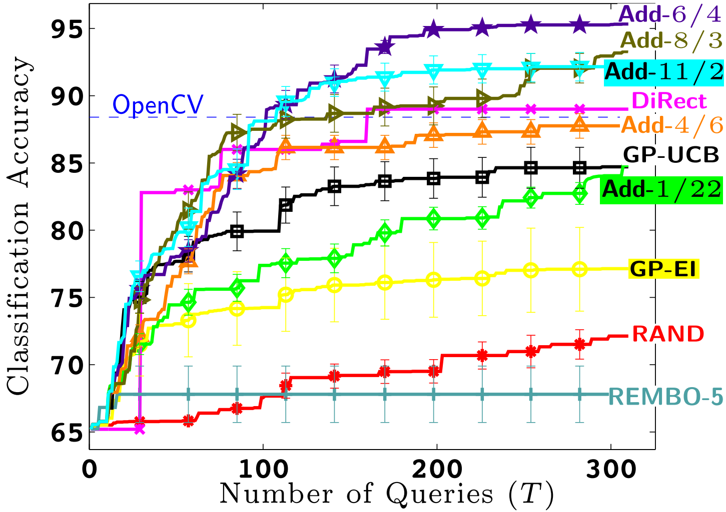
Viola & Jones Face Detection

A cascade of 22 weak classifiers.
Image classified negative if the score $<$ threshold at any stage.



- **Task:** Find optimal threshold values on a training set of 1000 images.
- 22 dimensions.
- Each query takes 30-40 seconds.
- Use 1000 DiRect evaluations to maximise acquisition function.

Viola & Jones Face Detection



Summary

- Additive assumption improves regret:
  exponential in $D \rightarrow$ *linear* in D.
- Acquisition function is easy to maximise.
- Even for non-additive f is not additive, **Add-GP-UCB** does well in practice.

Summary

- Additive assumption improves regret:
  exponential in $D \rightarrow$ *linear* in D.
- Acquisition function is easy to maximise.
- Even for non-additive f is not additive, **Add-GP-UCB** does well in practice.
- *Similar results hold for Matérn kernels and in bandit setting.*

Summary

- Additive assumption improves regret:
  exponential in $D \to$ *linear* in D.
- Acquisition function is easy to maximise.
- Even for non-additive f is not additive, **Add-GP-UCB** does well in practice.
- *Similar results hold for Matérn kernels and in bandit setting*.

Some open questions:

- How to choose (d, M)?
- Can we generalise to other acquisition functions?

Summary

- Additive assumption improves regret:
  exponential in $D \rightarrow$ *linear* in D.
- Acquisition function is easy to maximise.
- Even for non-additive f is not additive, **Add-GP-UCB** does well in practice.
- *Similar results hold for Matérn kernels and in bandit setting.*

Some open questions:

- How to choose (d, M)?
- Can we generalise to other acquisition functions?

Code available: `github.com/kirthevasank/add-gp-bandits`

Jeff's Talk: Friday 2pm @ Van Gogh

Thank You.