Gaussian Processes for Uncertainty Quantification
Part II

Javier González
Amazon Cambridge, UK

MLSS, Buenos Aires, Argentina

19th Jun 2018

Outline of the lecture

- Part I: *Introduction to Gaussian processes*
  - Basic description of Gaussian processes.
  - Gaussian processes with non Gaussian likelihoods.
  - Functional point of view on Gaussian processes and connections.
  - Deep Gaussian processes.
- Part II: **Decision making under uncertainty**
  - General framework for decision making.
  - Bayesian optimization.
  - Bayesian quadrature.
  - Experimental design.

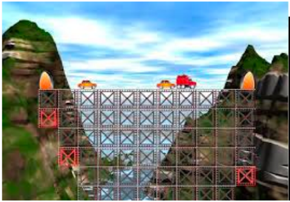
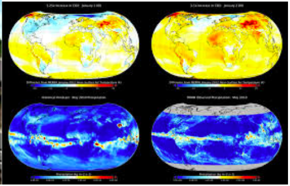
The world is an uncertain place

Uncertainty quantification

UQ is the science of quantitative characterization and reduction of uncertainties in both computational and real world applications (Wikipedia).

- Characterization: Gaussian process, other probabilistic models
- Reduction?

Simulations

Simulators are great but:

- Are often slow and expensive to run.
- Can only simulate just what it has been programmed to simulate.
- Simulators are black boxes hard to interpret.

Basic idea of surrogate modelling/emulation

[O'Hagan 2013; O' Hagan, 2006; Conti and O'Hagan, 2010]

Replace (or complement) the simulator with and emulator.

Emulator: probabilistic model fitted on simulation runs.

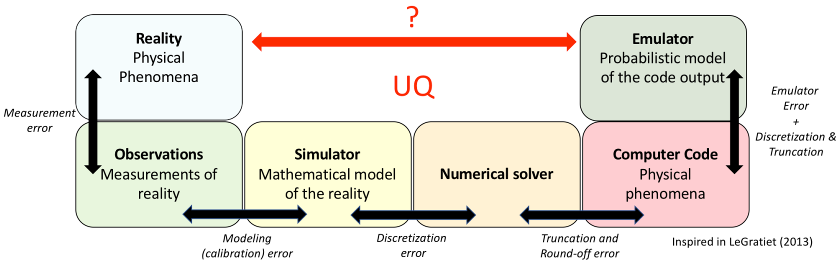
- Predictions are inexpensive.
- Predictions come with a level of uncertainty (GP emulators).

An emulator is a 'model of a model'

Areas of interest in uncertainty quantification

1. **Statistical emulation of complex simulators.**
   - Scalable UQ.
   - Differentially enhanced UQ.
2. **Systems understanding.**
   - Sensitivity analysis.
   - Propagation of uncertainty.
3. **Uncertainty in the loop.**
   - Reinforcement Learning.
   - Experimental design.
   - Probabilistic numerics (quadrature, optimization, etc.).

Uncertainty propagation in emulation



UQ deals with the end-to-end study of the impact of all forms of error and uncertainty in the models that we use to analyse or build a system of interest.

Uncertainty propagation in complex pipelines

1. **Model all sources of uncertainty.**

History of semi-mechanistic models in UQ

Risk models for catastrophes insurance were done purely in statistical fashion

Pure statistical models

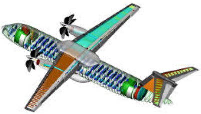
Incorporating physics or human behavior to improve predictions, ex. physics-based model of water drainage to assess potential damage from rainfall.

Mid 1990s

Semi mechanistic models

Deterministic engineering models were used to build complex physical systems

Pure mechanistic models



Incorporating element of uncertainty to account for lack of knowledge, ex. important physical parameters, randomness in operating circumstances, ignorance about the form of a 'correct' model.

Semi-mechanistic emulators -> Decisions under uncertainty

1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**

Decisions under uncertainty

Statistical inference:

$$\textit{model} + \textit{data} \rightarrow \textit{prediction}$$

- We have learned how to do this with Gaussian processes.
- GPs but not the only way: Bayesian neural networks, etc.
- Machine learning promises automatic decision making.

Decision making:

$$\textit{Predictions} \rightarrow \textit{Decisions}$$

- The models we use need to tell us when they don't know.
- We need probabilistic models in decision making (as GPs).

Decisions under uncertainty

Statistical inference:

$$\textit{model} + \textit{data} \rightarrow \textit{prediction}$$

- We have learned how to do this with Gaussian processes.
- GPs but not the only way: Bayesian neural networks, etc.
- Machine learning promises automatic decision making.

Decision making:

$$\textit{Predictions} \rightarrow \textit{Decisions}$$

- The models we use need to tell us when they don't know.
- We need probabilistic models in decision making (as GPs).

Decisions under uncertainty

Inference

- *Things that I know:*

$$y$$

- *Things that I don't know:*

$$y*$$

- *Description of the world:*

$$p(y^*, y)$$

- *What I need:*

$$p(y^*|y)$$

Decisions

- *Actions I can take:*

$$a \in \mathcal{A}$$

- *Reward I gain:*

$$R(a|y, y^*)$$

- *'Optimal' decision*:

$$a^* = arg \max_{\mathcal{A}} \alpha(a; R, p)$$

Example:

$$\alpha(a; R, p) = \mathbb{E}_p R(a|y, y^*)$$

Decisions under uncertainty

Inference

- *Things that I know:*

$$y$$

- *Things that I don't know:*

$$y*$$

- *Description of the world:*

$$p(y^*, y)$$

- *What I need:*

$$p(y^*|y)$$

Decisions

- *Actions I can take:*

$$a \in \mathcal{A}$$

- *Reward I gain:*

$$R(a|y, y^*)$$

- *'Optimal' decision*:

$$a^* = arg\max_{\mathcal{A}} \alpha(a; R, p)$$

Example:

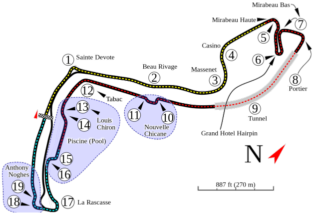
$$\alpha(a; R, p) = \mathbb{E}_p R(a|y, y^*)$$

1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**
3. **Decision making under uncertainty requires a model of the unknowns and a decision function.**

Uncertainty in decision making. A F1 example

Before the race:

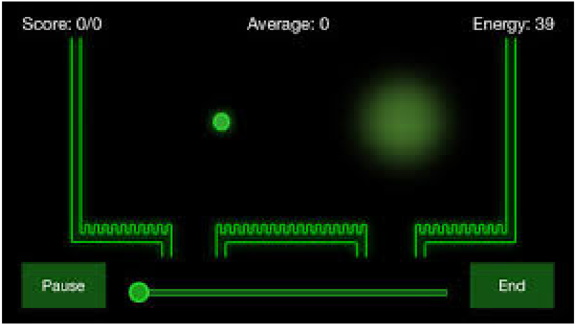
- F1 teams use simulations to define the strategy.
- Expensive, cannot be used in real time.
- Replace simulator with an emulator (model fitted in simulations).



During the race:

- Emulator for quick decisions.
- *If uncertainty is low*: go ahead.
- *If uncertainty is large*: run simulation.

Uncertainty in decision making. Kappenball

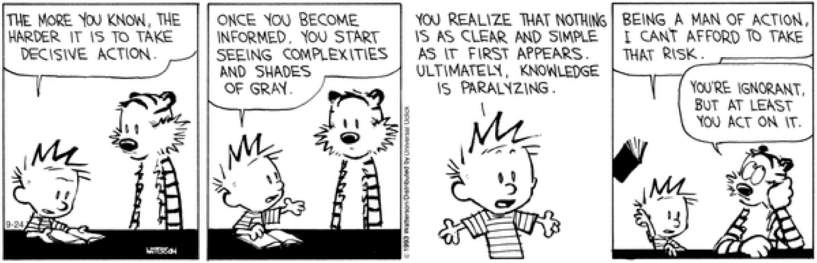


The uncertainty of the environment is key in optimal decision making

Goals of this lecture

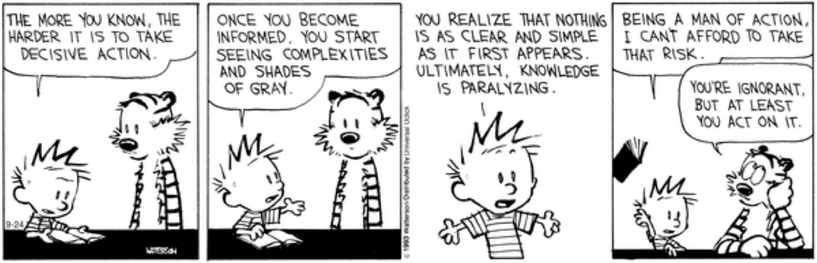
- Motivate and analyse different scenarios that lead to different choices of $\alpha(a; R, p)$.
- Focus of optimization, quadrature and experimental design (UQ and probabilistic numerics).
- Reinforcement learning is another interested case. We are only covering briefly today.

In essence...



We will learn how to *act on our ignorance* when making decisions

In essence...



We will learn how to *act on our ignorance* when making decisions

Elements when making a decision

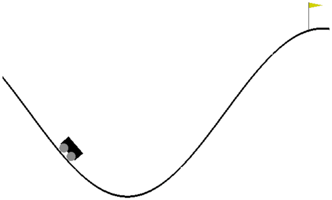
We may want to make different types of decisions.

We need to know:

- **Environment**, $p(y)$: **where** are we making the decision.
- **Actions set,** $\mathcal{A}$: **what** can we do.
- **Reward function,** R: **why** we are making a decision.
- **Policy,** $\alpha(a; R, p)$: **how** we make the decision.

Reinforcement learning

Goal: define a sequence of actions (push right or left) to reach the flag in T steps.



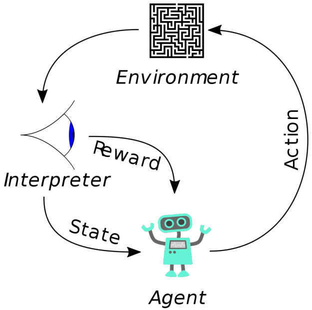
$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{a}_t)$$

- $\mathbf{x}_t = (p_t, v_t)$: position and velocity of the car at time t.
- $\mathbf{a}_t$ action force at time t.
- $\mathbf{u}_t = \pi(\mathbf{x}_t, \theta)$ is the policy, for instance:

$$\pi(\mathbf{x}, \theta) = \theta_0 + \theta_p p + \theta_v v.$$

Reinforcement learning

Canonical loop for decisions of an automatic agent



While *more actions*:

1. Observe the environment.
2. Update our state (model).
3. Make an action.

Related problems

That can also be solved using the same type of loop

- Optimization:

$$x^* = \arg\min_{\mathcal{X}} f(x).$$

- Quadrature:

$$Z = \int_{\mathcal{X}} f(x)p(x)dx.$$

Common framework

Active learning, Bayesian optimization, bandits, reinforcement, etc, all have a common ground:

- Use some form of belief of the environment.
- Sequential decisions using some form of $\alpha(a; R, p)$.
- Decisions influence rewards.
- Described as 'Exploration/Exploitation' problems.

Exploration vs. exploitation

The exploration exploitation dilemma is present in most of our day-by-day decisions.

Bayesian reasoning.

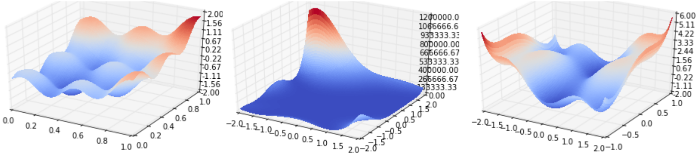
1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**
3. **Decision making under uncertainty requires a model of the unknowns and a decision function.**
4. **AL, BayesOpt, bandits, RL, share a common decision making framework.**

Bayesian optimization

Global optimization

Consider a 'well behaved' function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^D$ is a bounded domain.

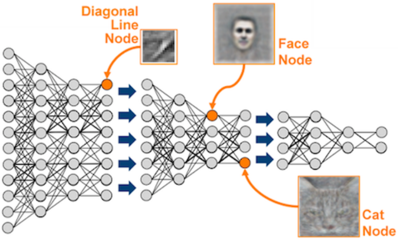
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



- f is explicitly unknown and multimodal.
- Evaluations of f may be perturbed.
- Evaluations of f are expensive.

Expensive functions, who doesn't have one?

Parameter tuning in ML algorithms.



- Number of layers/units per layer.
- Weight penalties, learning rates, etc.

Figure source: http://theanalyticsstore.com/deep-learning

Expensive functions, who doesn't have one?

Many other problems:

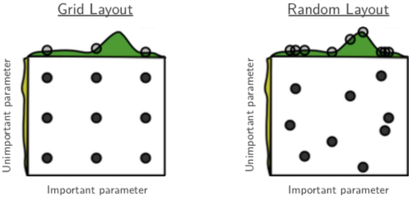
- Robotics, control, reinforcement learning.
- Scheduling, planning.
- Compilers, hardware, software.
- Industrial design.
- Intractable likelihoods.

What to do?

Option 1: Use previous knowledge

Option 2: Grid search?

Option 3: We can sample the space uniformly [Bergstra and Bengio 2012]



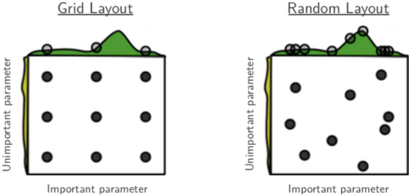
Option 4: Can we do better?

What to do?

Option 1: Use previous knowledge

Option 2: Grid search?

Option 3: We can sample the space uniformly [Bergstra and Bengio 2012]



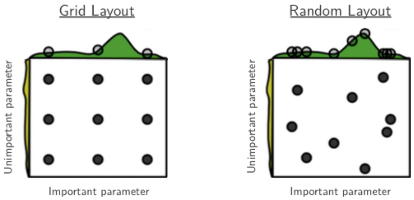
Option 4: Can we do better?

What to do?

Option 1: Use previous knowledge

Option 2: Grid search?

Option 3: We can sample the space uniformly [Bergstra and Bengio 2012]



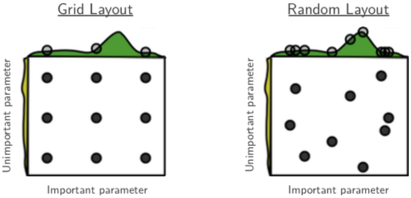
Option 4: Can we do better?

What to do?

Option 1: Use previous knowledge

Option 2: Grid search?

Option 3: We can sample the space uniformly [Bergstra and Bengio 2012]



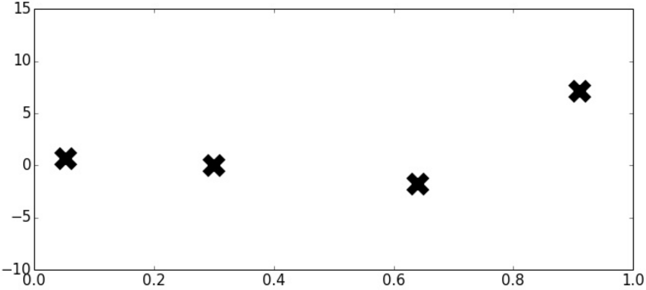
Option 4: Can we do better?

Problem (the audience is encouraged to participate!)

- Find the optimum of some function f in the interval [0,1].
- f is (L-Lipchitz) continuous and differentiable.
- Evaluations of f are exact and we have 4 of them!

Situation

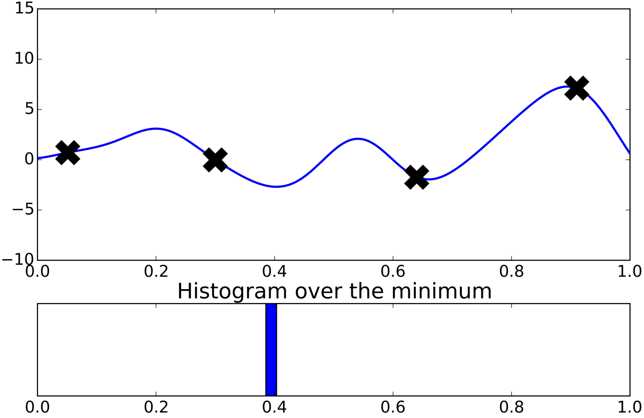
We have a few function evaluations



Where is the minimum of f?
Where should we take the next evaluation?

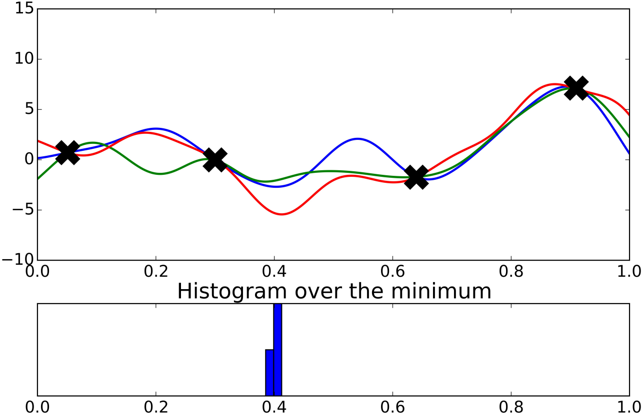
Intuitive solution

One curve



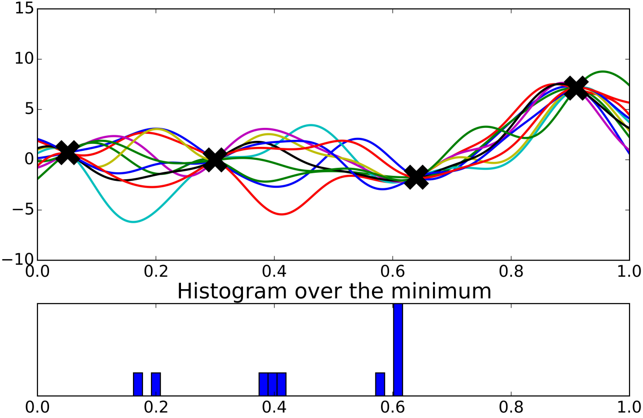
Intuitive solution

Three curves



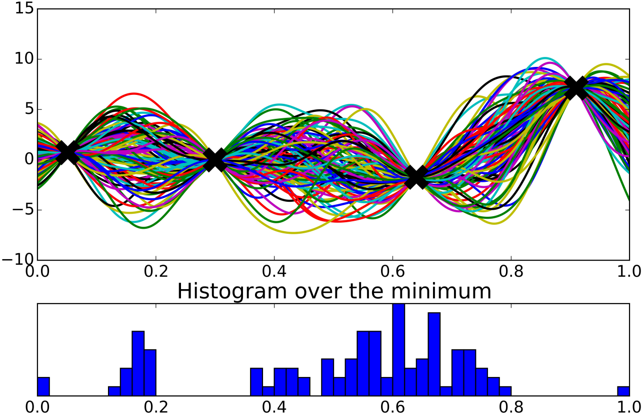
Intuitive solution

Ten curves



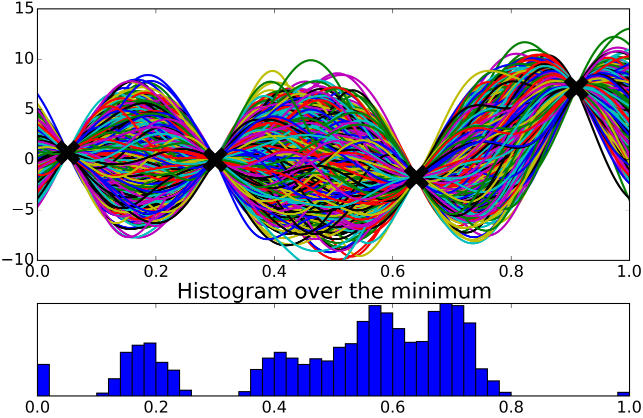
Intuitive solution

Hundred curves



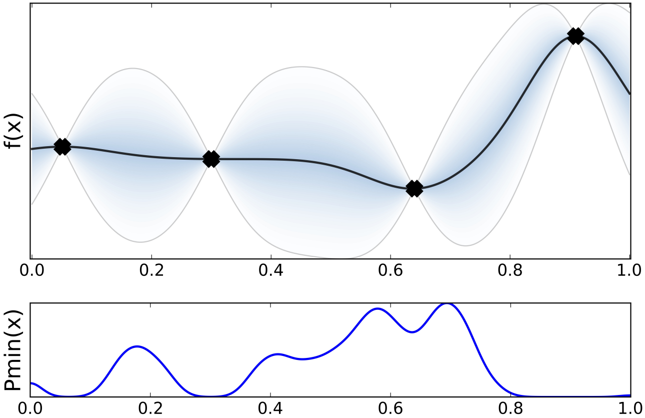
Intuitive solution

Many curves



Intuitive solution

Infinite curves



Surrogate modelling

1. Use a surrogate model of f to carry out the optimization.
2. Define an utility function to collect new data points satisfying some optimality criterion: *optimization* as *decision*.
3. Study *decision* problems as *inference* using the surrogate model: use a probabilistic model able to calibrate both, epistemic and aleatoric uncertainty.

Reward (regrets) in Bayesian optimization

Minimize the loss in a sequence $x_1, \ldots, x_n$

1. Cumulative regret

$$r_N = \sum_{n=1}^{N} f(x_n) - Nf(x_M)$$

2. Final regret

$$r_N = f(x_n) - Nf(x_M)$$

Bayesian optimization

Find

$$x^* = \arg\min_{\mathcal{X}} f(x).$$

- **Environment**: Gaussian process on the objective, $p(f)$.
- **Actions set, $\mathcal{A}$**: Space $\mathcal{X}$ where f is evaluated.
- **Reward function, R**: Minus the cumulative/final regret.
- **Policy, $\alpha(a; R, p)$** : **??**

Bayesian optimization

While *more actions*:

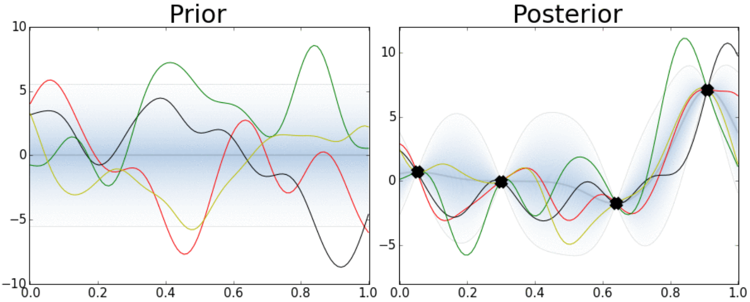
1. Observe the environment.
2. Update our state (model).
3. Make an action.

- **Environment**: Gaussian process on the objective, $p(f)$.
- **Actions set, $\mathcal{A}$**: Space $\mathcal{X}$ where f is evaluated.
- **Reward function, R**: Minus the cumulative/final regret.
- **Policy, $\alpha(a; R, p)$** : **??**

Surrogate model: Gaussian process

Default Choice: Gaussian processes [Rasmunsen and Williams, 2006]

Infinite-dimensional probability density, such that each linear finite-dimensional restriction is multivariate Gaussian.



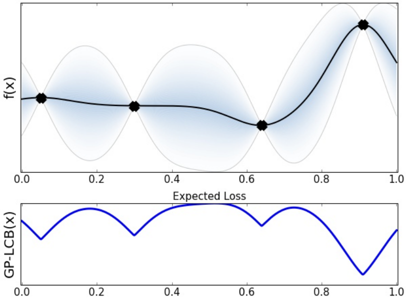
- Model $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$ is determined by the *mean function* $m(x)$ and *covariance function* $k(x, x'; \theta)$.

GP upper (lower) confidence band

[Srinivas et al., 2010]

Direct balance between exploration and exploitation:

$$\alpha_{LCB}(\mathbf{x}; \theta, \mathcal{D}) = -\mu(\mathbf{x}; \theta, \mathcal{D}) + \beta_t \sigma(\mathbf{x}; \theta, \mathcal{D})$$



GP upper (lower) confidence band

[Srinivas et al., 2010]

- In noiseless cases, it is a lower bound of the function to minimize.
- This allows to compute a bound on how close we are to the minimum.
- Optimal choices available for the 'regularization parameter'.

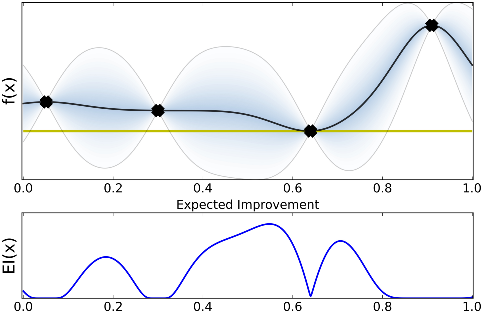
Theorem 1 *Let* $\delta \in (0,1)$ *and* $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$. *Running GP-UCB with* β_t *for a sample* f *of a GP with mean function zero and covariance function* $k(\boldsymbol{x}, \boldsymbol{x}')$, *we obtain a regret bound of* $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$ *with high probability. Precisely, with* $C_1 = 8/\log(1+\sigma^{-2})$ *we have*

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1 - \delta.$$

Expected improvement

[Jones et al., 1998]

$$\alpha_{EI}(\mathbf{x};\theta,\mathcal{D}) = \int_y \max(0, y_{best} - y)\, p(y|\mathbf{x};\theta,\mathcal{D})\, dy$$



Expected improvement

[Jones et al., 1998]

- Perhaps the most used acquisition.
- Explicit form available for Gaussian posteriors.
- It is too greedy in some problems. It is possible to make more explorative adding an 'explorative' parameter

$$\alpha_{EI}(\mathbf{x};\theta,\mathcal{D}) = \sigma(\mathbf{x};\theta,\mathcal{D})(\gamma(x)\Phi(\gamma(x))) + \mathcal{N}(\gamma(x);0,1).$$

where

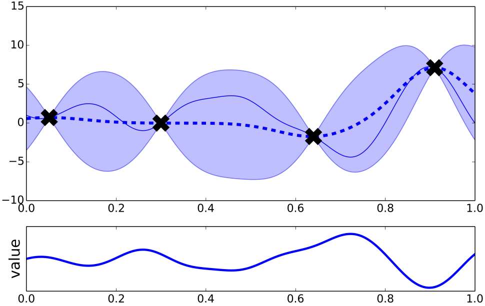
$$\gamma(x) = \frac{f(x_{best}) - \mu(\mathbf{x};\theta,\mathcal{D}) + \psi}{\sigma(\mathbf{x};\theta,\mathcal{D})}.$$

Thompson sampling

Probability matching [Rahimi and B. Recht, 2007]

$$\alpha_{THOMPSON}(\mathbf{x};\theta,\mathcal{D}) = g(\mathbf{x})$$

$g(\mathbf{x})$ is sampled form $\mathcal{GP}(\mu(x), k(x, x'))$



Thompson sampling

Probability matching [Rahimi and B. Recht, 2007]

- Getting samples of a GP at a finite set of locations is easy.
- More difficult is to generate ‘continuous’ samples.

Bochner’s lemma: existence of the Fourier dual of k, $s(\omega)$which is equal to the spectral density of k.

$$k(x, x') = \nu \mathbb{E}_{\omega} \left[e^{-i\omega^T(x-x')} \right] = 2\nu \mathbb{E}_{\omega, b} \left[\cos(\omega x^T + b) \cos(\omega x^T + b) \right]$$

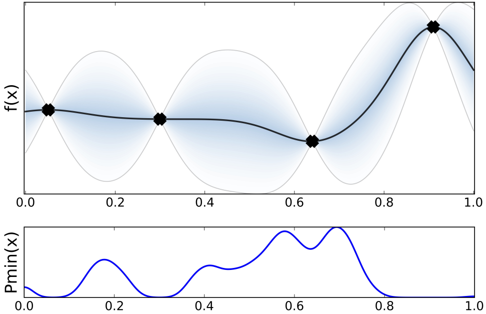
With sampling and this lemma (taking $p(w) = s(\omega)/\nu$ and $b \sim \mathcal{U}[0, 2\pi]$) we can construct a feature based approximation for sample paths of the GP.

$$k(x, x') \approx \frac{\nu}{m} \sum_{i=1}^{m} e^{-i\omega^{(i)T}x} e^{-i\omega^{(i)T}x'}$$

Information-theoretic approaches

[Hennig and Schuler, 2013; Hernández-Lobato et al., 2014]

$$\alpha_{ES}(\mathbf{x};\theta,\mathcal{D}) = H[p(x_{min}|\mathcal{D})] - \mathbb{E}_{p(y|\mathcal{D},\mathbf{x})}[H[p(x_{min}|\mathcal{D}\cup\{\mathbf{x},y\})]]$$



Information-theoretic approaches

[Hennig and Schuler, 2013; Hernández-Lobato et al., 2014]

Use the distribution of the minimum

$$p_{min}(x) \equiv p[x = \arg\min f(x)] = \int_{f:I\to\Re} p(f) \prod_{\substack{\tilde{x}\in I \\ \tilde{x}\neq x}} \theta[f(\tilde{x}) - f(x)]df$$

where θ is the Heaviside's step function. No closed form!

- Thompson sampling to approximate the distribution.
- Generate many sample paths from the GP.
- Optimize them to take samples from $p_{min}(x)$.

The choice of utility matters

[Hoffman, Shahriari and de Freitas, 2013]

The choice of the utility may change a lot the result of the optimisation.

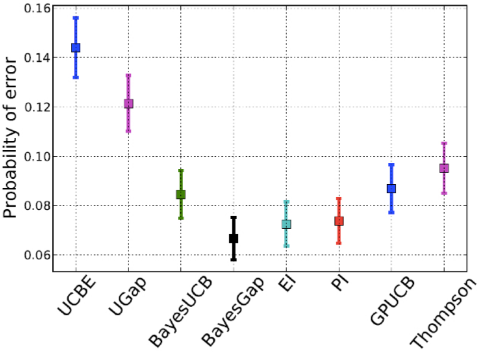


Illustration of BO

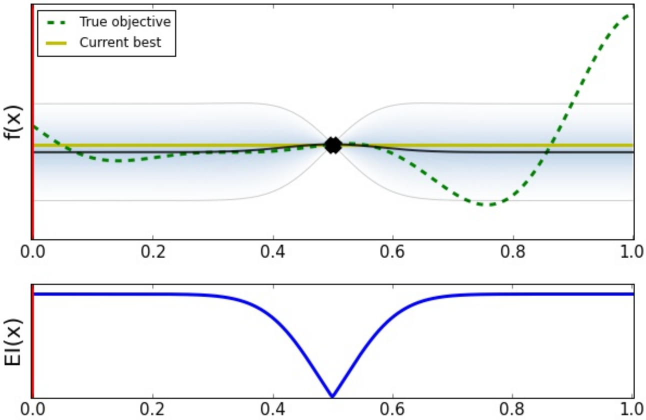


Illustration of BO

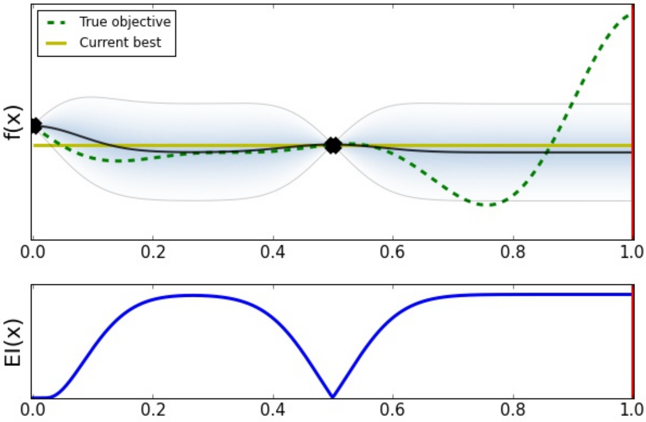


Illustration of BO

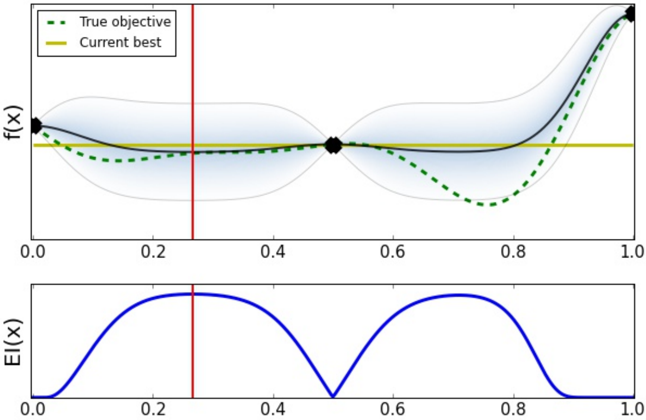


Illustration of BO

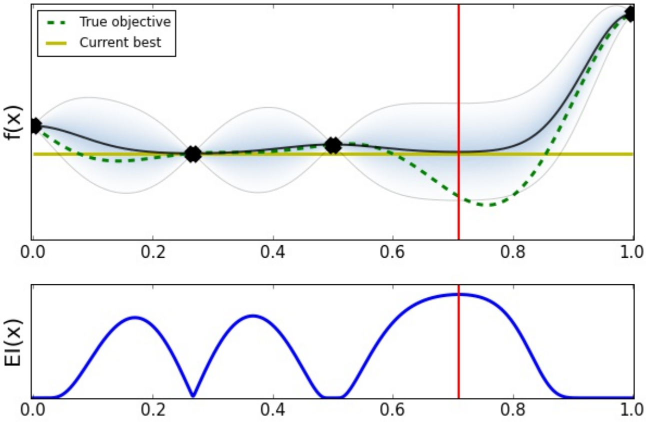


Illustration of BO

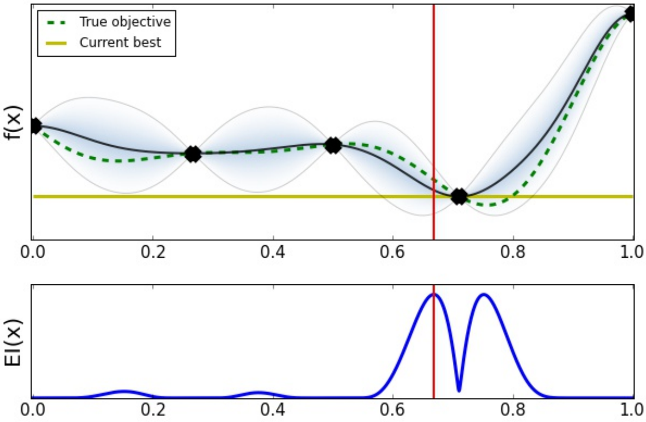


Illustration of BO

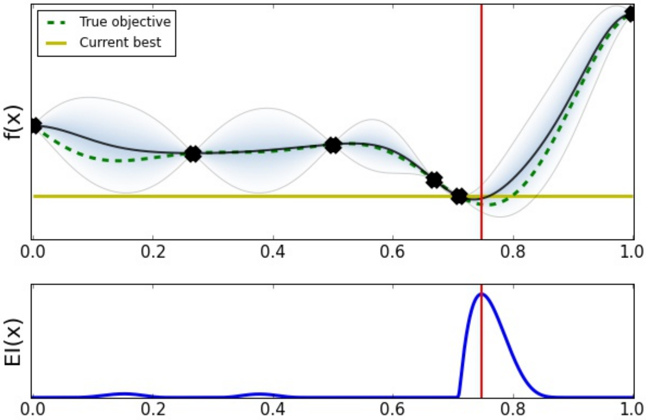


Illustration of BO

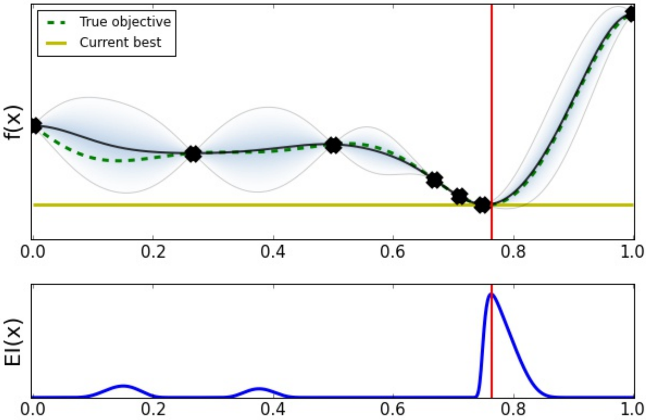
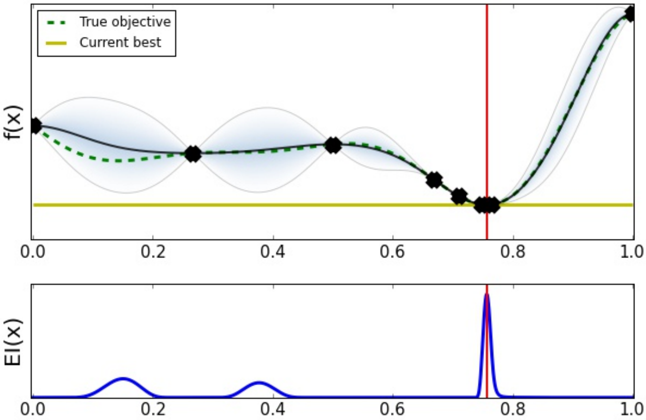


Illustration of BO



Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.

- In high dimensions the search becomes uninformative. **Solution**: Parallel approaches, informative priors.

- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.

- Limitations in the updates of the GP. **Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.

- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.

- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.

- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Limitations of Bayesian optimization

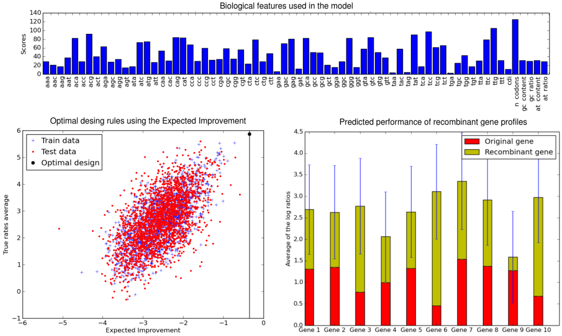
- Optimizing the acquisition may be hard. **Solution**: Multiple local optimizers.
- In high dimensions the search becomes uninformative.**Solution**: Parallel approaches, informative priors.
- Structured inputs/conditional parameters can be hard to handle. **Solution**: Latent variable surrogates, structured kernels.
- Limitations in the updates of the GP.**Solution**: Sparse GPs, Bayesian NNs.

Despite these, BayesOpt has been successfully applied in many applications.

Synthetic gene design

[Gonzalez et al, 2015]

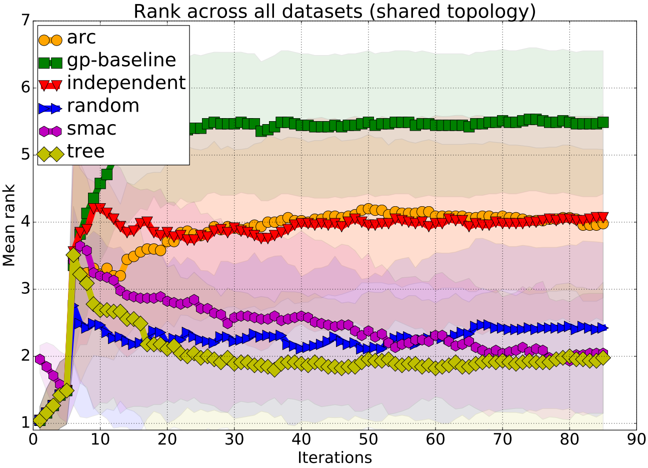
- Use mammalian cells to make protein products.
- Control the ability of the cell-factory to use synthetic DNA.



Optimize genes (ATTGGTUGA...) to best enable the cell-factory to operate most efficiently.

Optimization of neural networks

[Jenatton, Archembau, Gonzalez and Seeger, 2017]

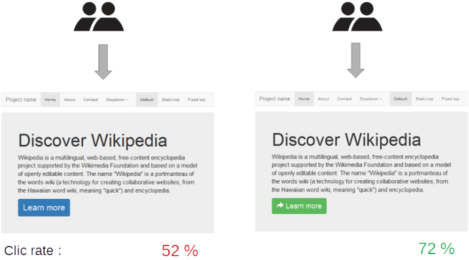


Raking of several BayesOpt algorithms used to configure a feed forward neural network on 50 datasets of the SVM light repository.

Preferential Bayesian optimization

[Gonzalez, Dai, Damianou and Lawrence, 2017]

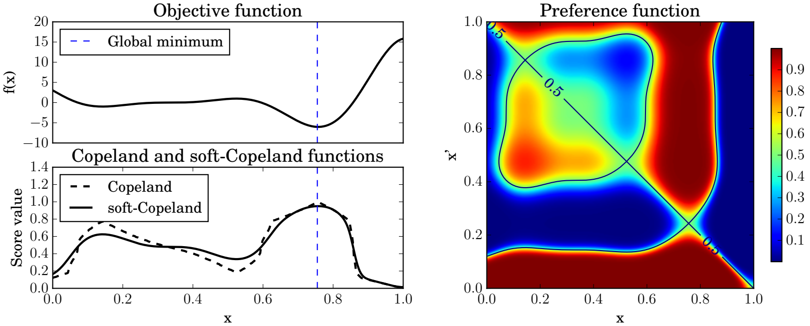
- The objective function of many tasks are difficult to precisely summarize into a single value.
- Comparison is almost always easier than rating for humans.
- Such observation has been exploited in A/B testing.



Idea

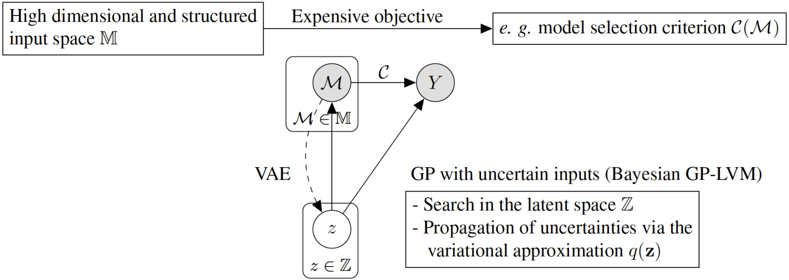
[Gonzalez, Dai, Damianou and Lawrence, 2017]

- To find the minimum of a latent function $g(x), x \in \mathcal{X}$.
- Observe only whether $g(\mathbf{x}) < g(\mathbf{x}')$ or not, for a *duel* $[\mathbf{x}, \mathbf{x}'] \in \mathcal{X} \times \mathcal{X}$.
- The outcomes are binary: *true* or *false*.
- Model the winner of duels with a Gaussian process for classifcation and learn a preference function.



Structured Variationally auto-encoded optimization

[Lu, Gonzalez, Dai and Lawrence, 2018]



Application: Image understanding

[Lu, Gonzalez, Dai and Lawrence, 2018]

Use Structured Bayesian optimization to search for an XML configuration of the "Minecraft" engine to reproduce three target images

Review articles to go further

A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning
Brochu, E.; Cora, V. M. & De Freitas, N.
Preprint arXiv:1012.2599, 2010

Taking the human out of the loop: A review of Bayesian optimization
Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P. & de Freitas, N.
Proceedings of the IEEE, 2016, 104, 148-175

1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**
3. **Decision making under uncertainty requires a model of the unknowns and a decision function.**
4. **AL, BayesOpt, Bandits, RL, share a common decision making framework.**
5. **Global optimization can be solved with GPs. The exploration/exploitation balance is the key.**

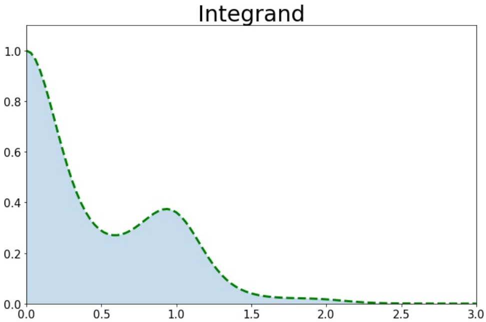
Bayesian quadrature

Introduction

Imagine that you need to compute

$$\int_0^3 f(x)dx = \int_0^3 \exp(-\sin(3x) - x^2)dx$$

and you cannot ask your old analysis teacher...

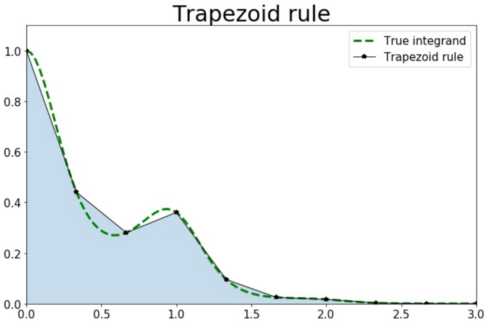


What can I do?

Cubature rules (or quadrature in 1D)

- Collect points in $x_1, \ldots, x_n$ in $[0, 3]$.

$$\int_0^3 f(x)dx \approx \sum_{i=1}^{n-1} w_i f(x_i)$$



How to select $x_1, \ldots, x_n$?

Several options:

- **Monte Carlo**: random samples in $[0, 3]$.
- **Quasi Monte Carlo**: pseudo random samples in $[0, 3]$.

Take $w_i = 1/N$:

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

Issues with this approaches:

- Don't use any of our knowledge about f (in principle).
- Don't give any idea of when to stop sampling.

Problem definition

In general, we want to estimate an integral

$$Z = \int_{\mathcal{X}} f(x)p(x)dx.$$

We are interested in cases where:

- The primitive of f is unknown.
- Evaluations of f are expensive.
- $p(x)$ is some measure of interest.

Applications

Most of what we do in the Bayesian world is an integral:

- **Moments**

$$Z = \mathbb{E}_p[f] = \int f(x)p(x)dx$$

- **Model evidence**

$$Z = p(y|X, \mathcal{D}) = \int p(y|X, \theta, \mathcal{M})p(\theta|\mathcal{M})d\theta$$

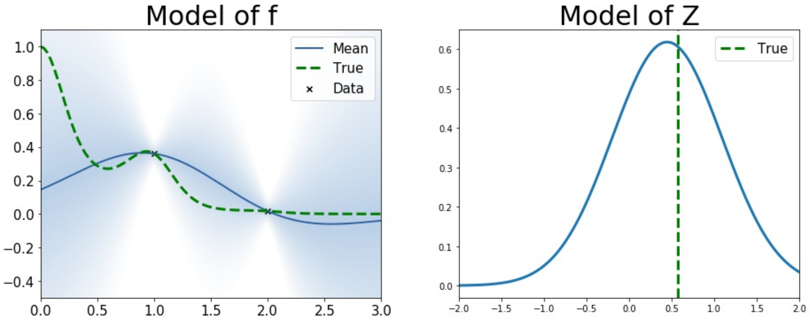
- **Predictions (marginalization)**

$$Z = p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \mathcal{D}, \theta)p(\theta|\mathcal{D})d\theta$$

Model based (Bayesian) quadrature

[Diaconis, 1988]

- Prior (Gaussian process) on the integrand f.
- The posterior over f induces a posterior over Z.



Issue: f is positive but that's not reflected in the model.

Inducing probability distribution over Z

- Integration is a linear operator.
- GPs are closed under linear operations.
- The integral of a GP is a GP (univariate Gaussian).

$$p\left(\int_{\mathcal{X}} f(x)dx\right) = \mathcal{N}\left(Z; \int_{\mathcal{X}} \mu(x)dx, \int_{\mathcal{X}} K(x,x')dxdx'\right)$$

UQ helps to compute integrals

Quantifying the uncertainty of the integral

- All the uncertainty has been 'pushed' to the integral.
- We know when we are close to a good estimate (assuming model is correct).
- Use it to collect new points to improve our estimate of Z.

$$\mathbb{E}[Z|\mathcal{D}] = \int \mu_{f|\mathcal{D}}(x)dx$$

$$\mathbb{V}ar(Z|\mathcal{D}) = \int\int_{\mathcal{X}} K_{f|\mathcal{D}}(x, x')dxdx'$$

Explicit form of the mean and variance of Z

- $X = \{x_i\}_{i=1}^n$.
- $\mathbf{f}$ the vector of components $\mathbf{f}_i = f(x_i)$.
- $\mathcal{GP}(0, k)$ fitted to the integrand f.
- $k_X(x) = (k(x, x_1), \ldots, k(x, x_1))^T$.

$$\mathbb{E}[Z|\mathcal{D}] = \left[\int k_X(x)dx\right] \mathbf{K}^{-1}\mathbf{f}$$

$$\mathbb{V}ar(Z|\mathcal{D}) = \int k(x, x')dxdx' - \left[\int k_X(x)dx\right] \mathbf{K}^{-1} \left[\int k_X(x)dx\right]^T$$

Two important considerations

1. BQ can be written in form of other quadrature rules for $\mathbf{w} = \left[\int k_X(x)dx\right]\mathbf{K}^{-1}$:

$$\mathbb{E}[Z|\mathcal{D}] = \left[\int k_X(x)dx\right]\mathbf{K}^{-1}\mathbf{f} = \mathbf{w}^T\mathbf{f} = \sum_i^n w_i^{BQ} f(x_i).$$

   Some kernels lead to known certain quadrature rules!

2. The quality of the approximation can be bounded by the norm of f in the RKHS induced by k.

$$|Z - \mathbb{E}[Z|\mathcal{D}]| \leq \|f\|_{\mathcal{H}} \|\mu - \hat{\mu}\|_{\mathcal{H}}$$

   where μ is the *kernel mean* and $\hat{\mu}$ is the *kernel mean* approximation in the RKHS induced by K.

Two important considerations

1. BQ can be written in form of other quadrature rules for $\mathbf{w} = \left[\int k_X(x)dx\right] \mathbf{K}^{-1}$:

$$\mathbb{E}[Z|\mathcal{D}] = \left[\int k_X(x)dx\right] \mathbf{K}^{-1}\mathbf{f} = \mathbf{w}^T\mathbf{f} = \sum_i^n w_i^{BQ} f(x_i).$$

   Some kernels lead to known certain quadrature rules!

2. The quality of the approximation can be bounded by the norm of f in the RKHS induced by k.

$$|Z - \mathbb{E}[Z|\mathcal{D}]| \leq \|f\|_{\mathcal{H}} \|\mu - \hat{\mu}\|_{\mathcal{H}}$$

   where μ is the *kernel mean* and $\hat{\mu}$ is the *kernel mean* approximation in the RKHS induced by K.

Bayesian quadrature

[FX Briol et al, 2015]

Find

$$Z = \int_{\mathcal{X}} f(x)p(x)dx$$

- **Environment**: Gaussian process on the integrand, $p(f)$.
- **Actions set, $\mathcal{A}$**: Space $\mathcal{X}$ where f is evaluated.
- **Reward function, R**: $|Z - \mathbb{E}[Z|\mathcal{D}]|$.
- **Policy, $\alpha(a; R, p)$** : **??**

Bayesian quadrature

[FX Briol et al, 2015]

While *more actions*:

1. Observe the environment.
2. Update our state (model).
3. Make an action.

- **Environment**: Gaussian process on the integrand, $p(f)$.
- **Actions set, $\mathcal{A}$**: Space $\mathcal{X}$ where f is evaluated.
- **Reward function, R**: $|Z - \mathbb{E}[Z|\mathcal{D}]|$.
- **Policy, $\alpha(a; R, p)$** : **??**

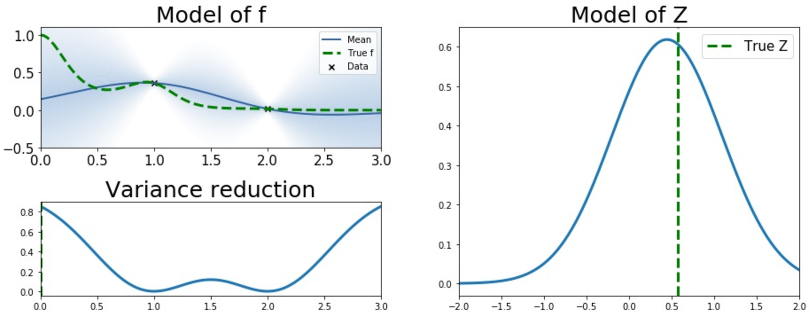
Policy for Bayesian quadrature

- Collect points where the information is more valuable.
- We can use the reduction in uncertainty of $Z|Data$.
- Optimal *off-line*: can collect for multiple points simultaneously.

$$\alpha(x^*) = \mathbb{V}ar(Z|\mathcal{D}) - \mathbb{E}_{p(y^*|x^*,\mathcal{D})}\left[\mathbb{V}ar(Z|\mathcal{D}\cup\{x^*,y^*\})|\mathcal{D},x^*\right]$$

Issue (or not?): $\alpha(x^*)$ does not depend on the values of y^*.

Illustration of Bayesian quadrature



Note that the estimate of the value of the integral is not bad, but we are uncertain about it.

Illustration of Bayesian quadrature

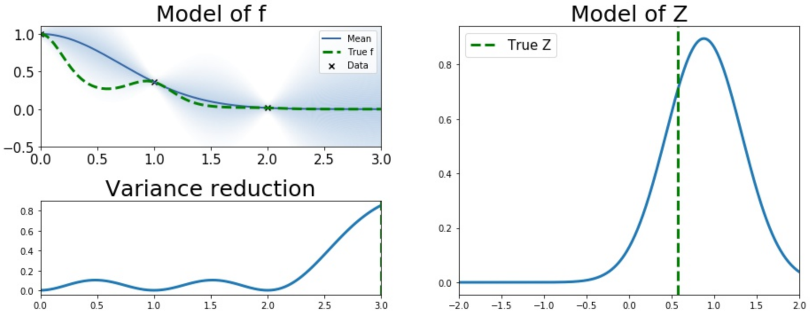


Illustration of Bayesian quadrature

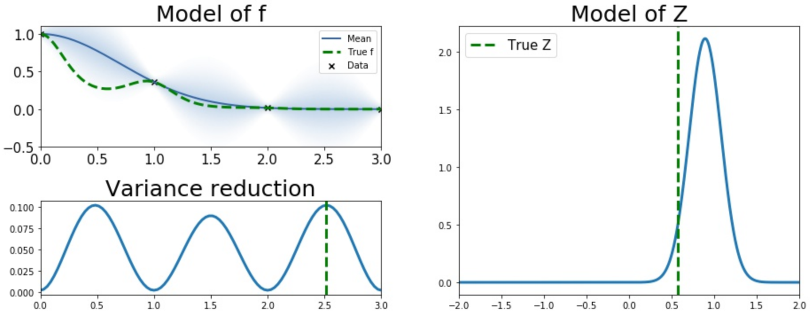


Illustration of Bayesian quadrature

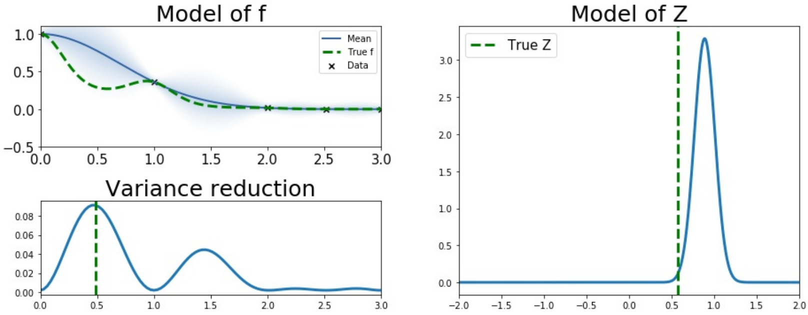


Illustration of Bayesian quadrature

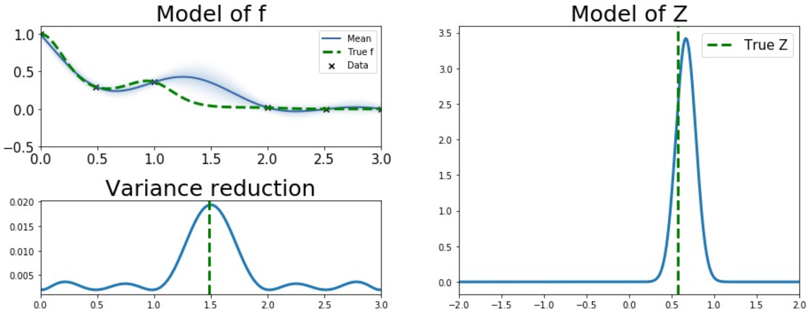


Illustration of Bayesian quadrature

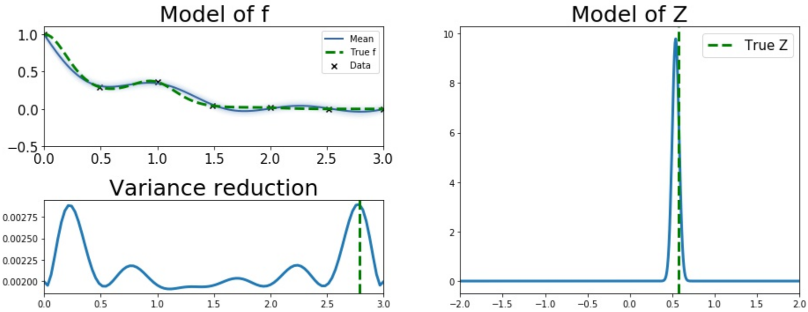


Illustration of Bayesian quadrature

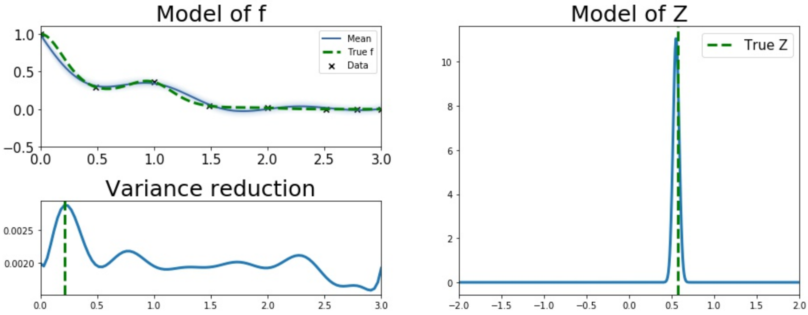
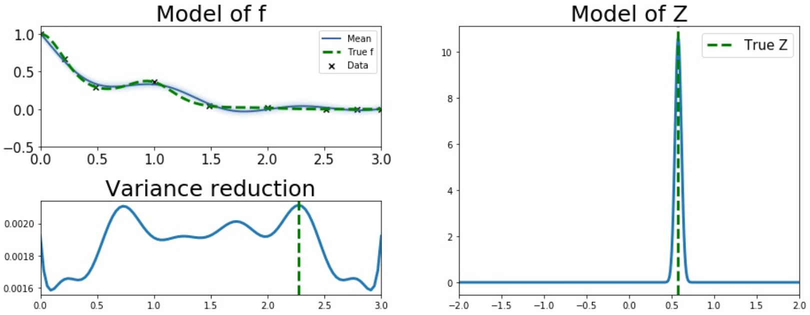


Illustration of Bayesian quadrature



Issues

[Osborne, et al. 2012; Gunter et al. 2014]

- **Positiveness**: Fits a GP to the $\log f$. Then fits a GP to the exponentiated $\log f$.
- **Function values**: Transform using the square function and use uncertainty sampling on the pulled-back GP.

$$\mathbf{x}_t = \arg\max_{\mathcal{X}} \mu(\mathbf{x})^2 K(\mathbf{x}, \mathbf{x})$$

Other limitations:

- Still does not work in high dimensions.
- Which model to use is a key question: we need global knowledge of f.

Review articles to go further

Probabilistic Integration: A Role for Statisticians in Numerical Analysis?
Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D.
Preprint ArXiv:1512.00933, 2015

On the relation between Gaussian process quadratures and sigma-point methods
Srkk, S., Hartikainen, J., Svensson, L., & Sandblom, F.
ArXiv Preprint Stat.ME 1504.05994, 2015

1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**
3. **Decision making under uncertainty requires a model of the unknowns and a decision function.**
4. **AL, BayesOpt, Bandits, RL, share a common decision making framework.**
5. **Global optimization can be solved with GPs. The exploration/exploitation balance is the key.**
6. **Quadrature problems can be solved with GPs. Although some issues remain, there are ways to tackle them.**

Experimental design

Core question

Given:

- A mapping function $y = f(x)$, expensive simulator for instance.
- A class of models to obtain $\hat{f}$ or $p(f)$.
- An algorithm to fit those models to inputs and outputs of f.

How to select $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$ so we can guarantee that the model/approximation is 'good'?

Experimental design

- **Model free**: Latin hypercubes, Sobol sequences, grids, etc.
- **Model based**: Collected points that maximize the information gain with respect to the model.

Latin design

Example of model free experimental design

$n \times n$ array filled with n different symbols, each occurring exactly once in each row and exactly once in each column.

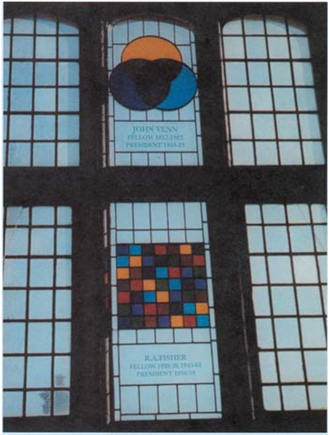
A	B	F	C	E	D
B	C	A	D	F	E
C	D	B	E	A	F
D	E	C	F	B	A
E	F	D	A	C	B
F	A	E	B	D	C

High discrepancy in the samples reduces variance.

Latin design

Example of model free experimental design

Window honors Ronald Fisher. Fisher's student, A. W. F. Edwards, designed this window for Caius College, Cambridge.



Using a GP to design an experiment

Model to use:

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

- ϵ_i is zero-mean Gaussian with variance σ^2.
- $p(f)$ a GP with some covariance k.
- We are interested on modelling the behaviour of f in $\mathcal{X}$.

How to get a sample of points $\mathcal{S}$ so we can estimate the function globally well?

How can we learn about f as rapidly as possible?

[Srinivas et al., 2010]

Bayesian experimental design:

- Informativeness of a set of points $\mathcal{S} \in \mathcal{X}$ is measured by the information collected.
- Mutual information between f and $\mathbf{y}_{\mathcal{S}} = \mathbf{f}_{\mathcal{S}} + \epsilon_{\mathcal{S}}$.
- Can be computed as: $I(\mathbf{y}_{\mathcal{S}}; f) = \frac{1}{2} \log |\mathbf{I} + \sigma^{-2}\mathbf{K}_{\mathcal{S}}|$.

Issue when using the mutual information

Finding $\mathcal{S}$ using the MI is NP-hard

- Approximates greedy search. Collect points in $\mathcal{S}$ iteratively:

$$\mathbf{x}_t = \arg\max_{\mathcal{X}} I(\mathbf{y}_{\mathcal{S}_{t-1}\bigcup\{\mathbf{x}\}}; f).$$

- This is equivalent to collect

$$\mathbf{x}_t = \arg\max_{\mathcal{X}} \sigma_{t-1}(\mathbf{x}).$$

Issue when using the mutual information

Finding $\mathcal{S}$ using the MI is NP-hard

- Approximates greedy search. Collect points in $\mathcal{S}$ iteratively:

$$\mathbf{x}_t = \arg\max_{\mathcal{X}} I(\mathbf{y}_{\mathcal{S}_{t-1} \bigcup \{\mathbf{x}\}}; f).$$

- This is equivalent to collect

$$\mathbf{x}_t = \arg\max_{\mathcal{X}} \sigma_{t-1}(\mathbf{x}).$$

Other alternatives to experimental design

Integrated Variance, [Gorodetsky and Marzouk, 2016]

- Select the point that reduce the most the 'accumulated' variance in the entire domain $\mathcal{X}$.
- Equivalent to an expected integrated squared error of the posterior mean.

Select $\mathcal{S} = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*\}$ such that

$$\mathcal{S} = \arg\min_{\mathcal{X}} \int_{\mathcal{X}} \sigma(\mathbf{x}|\mathcal{S}) d\mathbf{x} \approx \frac{1}{N_{mc}} \sum_{i=1}^{N} \sigma(\mathbf{x}_i|\mathcal{S})$$

for N_{mc} is the number of Monte Carlo samples.

Other alternatives to experimental design

Integrated Variance, [Gorodetsky and Marzouk, 2016]

- Select the point that reduce the most the 'accumulated' variance in the entire domain $\mathcal{X}$.
- Equivalent to an expected integrated squared error of the posterior mean.

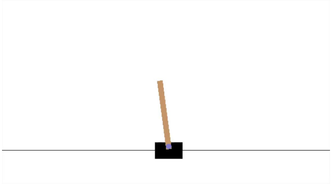
Select $\mathcal{S} = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*\}$ such that

$$\mathcal{S} = \arg\min_{\mathcal{X}} \int_{\mathcal{X}} \sigma(\mathbf{x}|\mathcal{S}) d\mathbf{x} \approx \frac{1}{N_{mc}} \sum_{i=1}^{N} \sigma(\mathbf{x}_i|\mathcal{S})$$

for N_{mc} is the number of Monte Carlo samples.

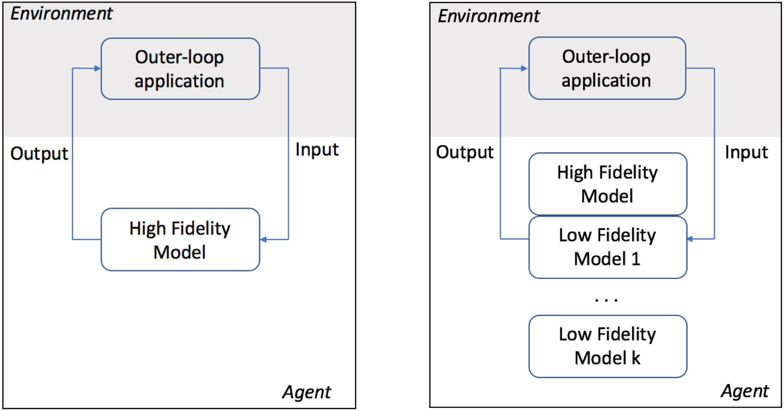
Multi-fidelity methods in experimental design

Combine data of different fidelities (qualities) in the same model:



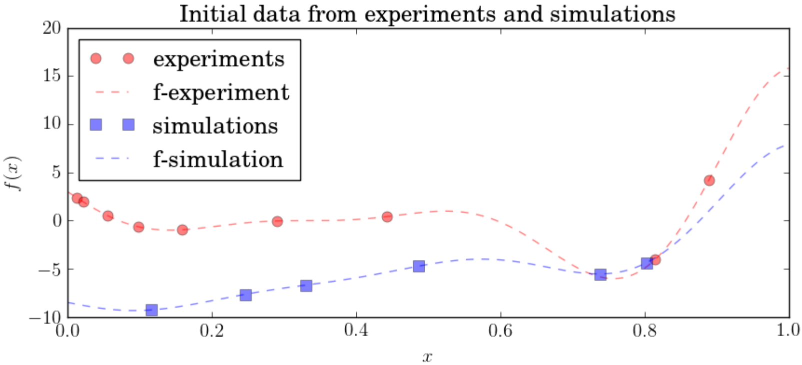
- Linear multifidelity model: $f_{high}(x) = \rho f_{low}(x) + \delta(x)$.
- The high fidelity is a GP so all experimental design ideas apply.

Multi-fidelity methods in experimental design



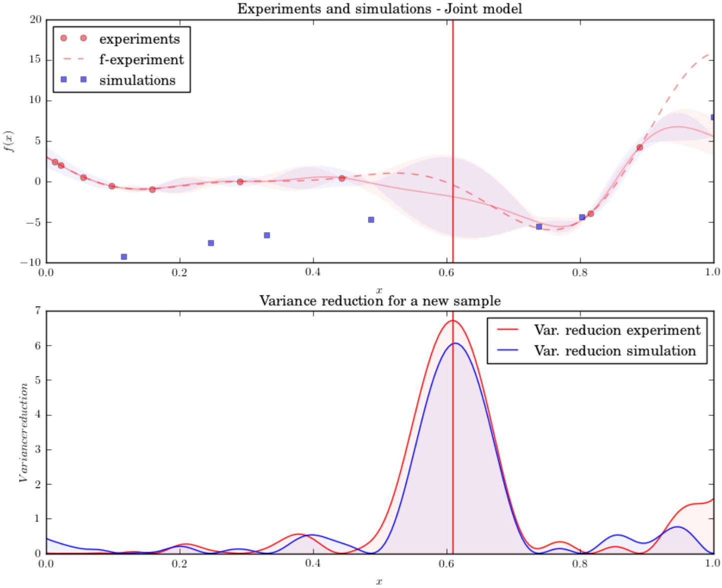
Single fidelity vs. multiple fidelities

Illustration

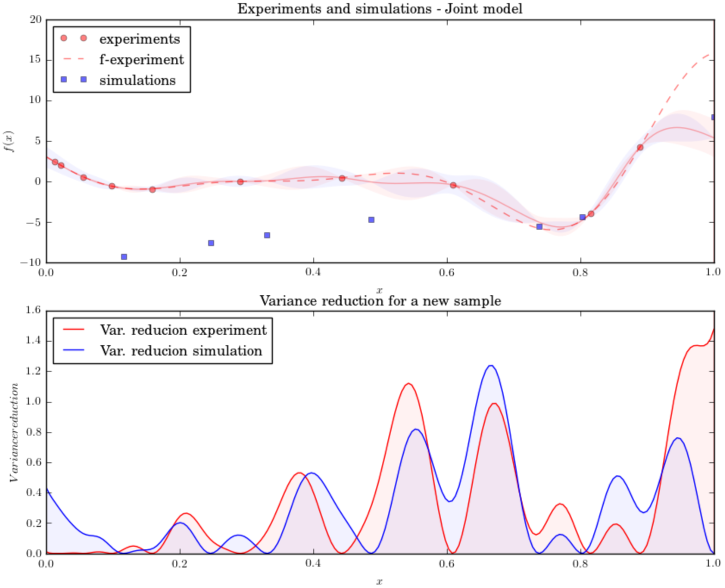


- Cost per simulation: 1u.
- Coste per experiment: 5u.

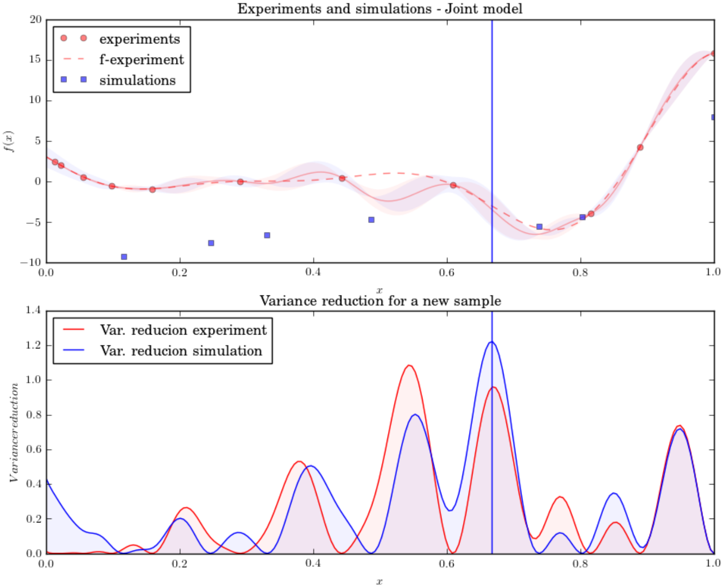
Example multi-fidelity experimental design



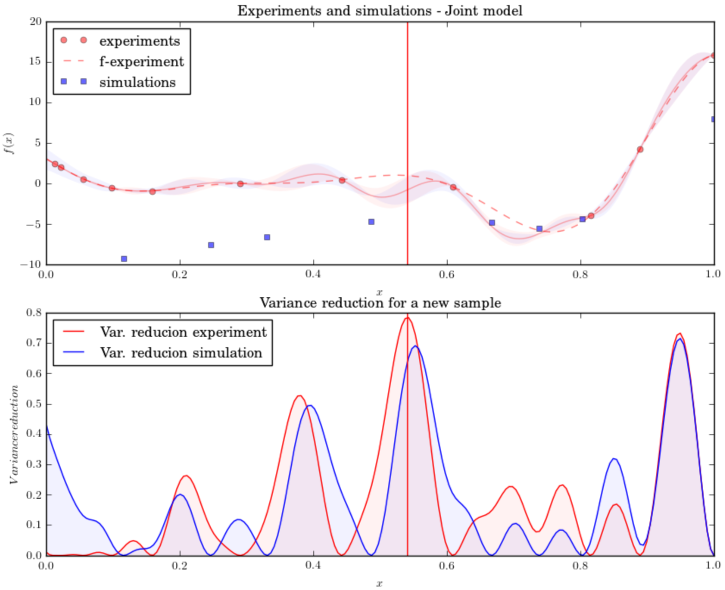
Example multi-fidelity experimental design



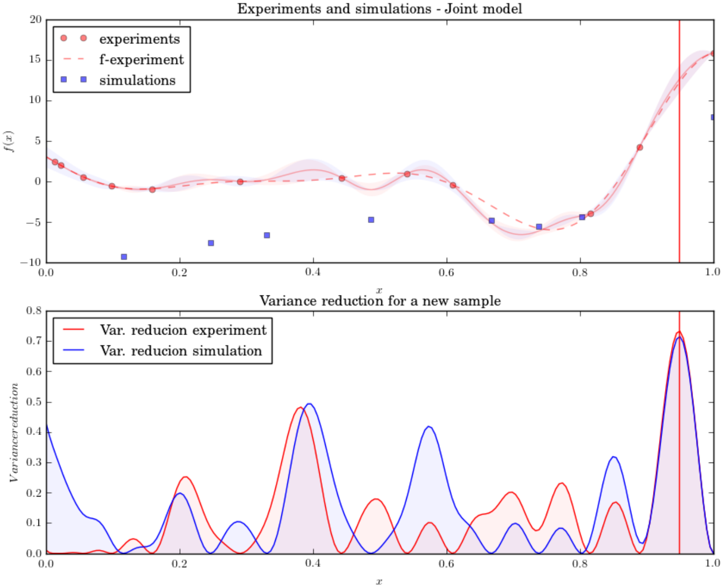
Example multi-fidelity experimental design



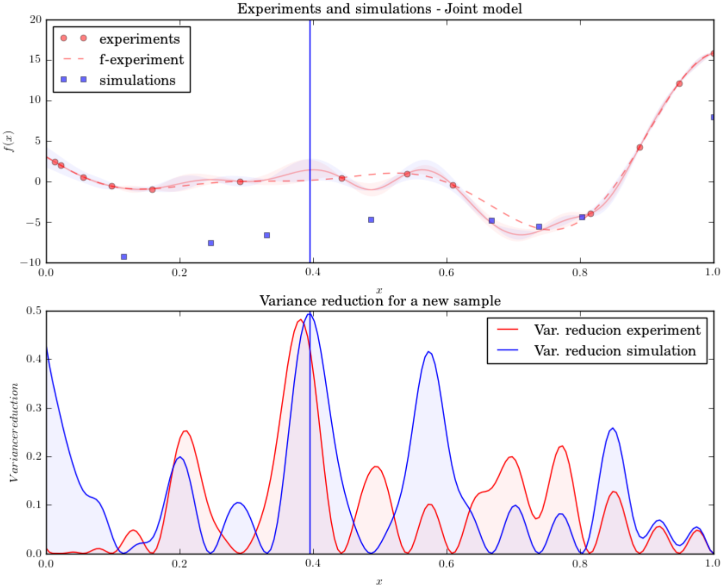
Example multi-fidelity experimental design



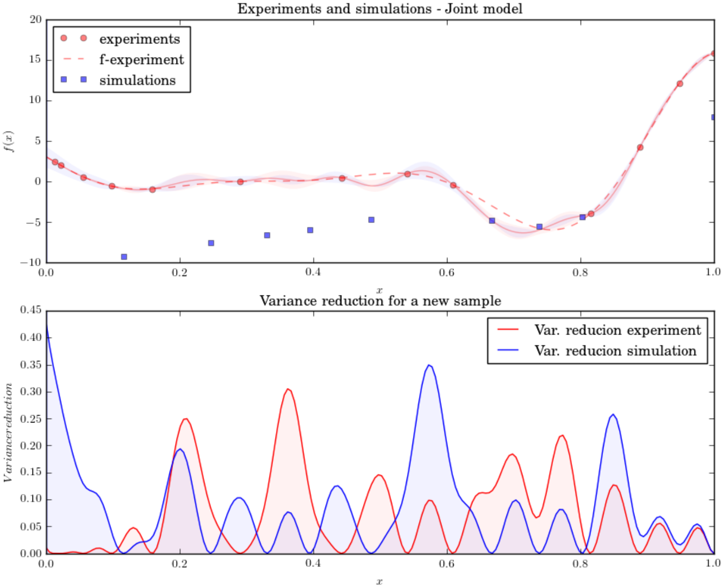
Example multi-fidelity experimental design



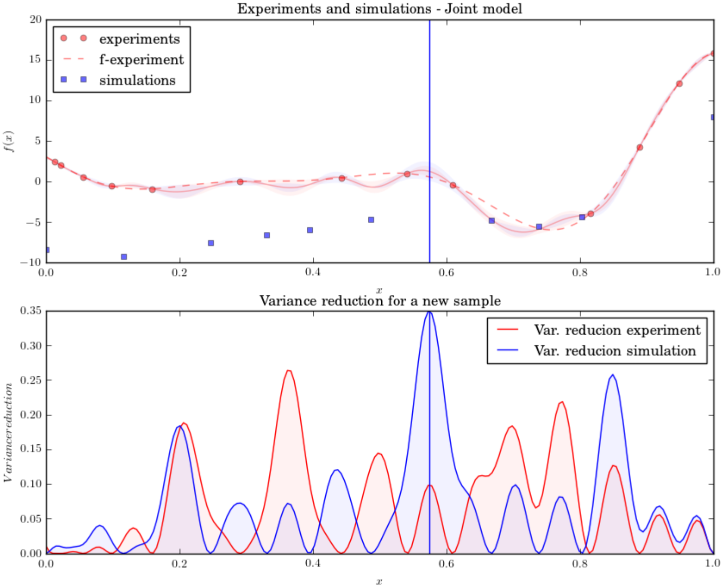
Example multi-fidelity experimental design



Example multi-fidelity experimental design



Example multi-fidelity experimental design



Review articles to go further

Bayesian Experimental Design: A Review
Kathryn Chaloner and Isabella Verdinelli
Statistical Science,Volume 10, Number 3 (1995), 273-304.

On a measure of information provided by an experiment
Lindley, D. V.
Annals of Mathematical Statistics, 27 (4): 9861005,1956

1. **Model all sources of uncertainty.**
2. **Use everything you know. Talk to the expert.**
3. **Decision making under uncertainty requires a model of the unknowns and a decision function.**
4. **AL, BayesOpt, Bandits, RL, share a common decision making framework.**
5. **Global optimization can be solved with GPs. The exploration/exploitation balance is the key.**
6. **Quadrature problems can be solved with GPs. Although some issues remain, there are ways to tackle them.**
7. **GPs are useful for experimental design. Multi-fidelity models give a framework for transfer learning.**

GPSS: Gaussian Process Summer School

- `http://ml.dcs.shef.ac.uk/gpss/`
- Next one is in Sheffield in September 2018.

Many thanks to:

Neil Lawrence, Zhenwen Dai, Andreas Damianou, Xiaoyu Lu, Mark Pullin, Andrei Paleyes, Maren Mahsereci, Vicky Scheider and Cliff McCollum.