Caltech

Bayesian Optimization

Danni Ma, Dimitar Ho

Table of Contents

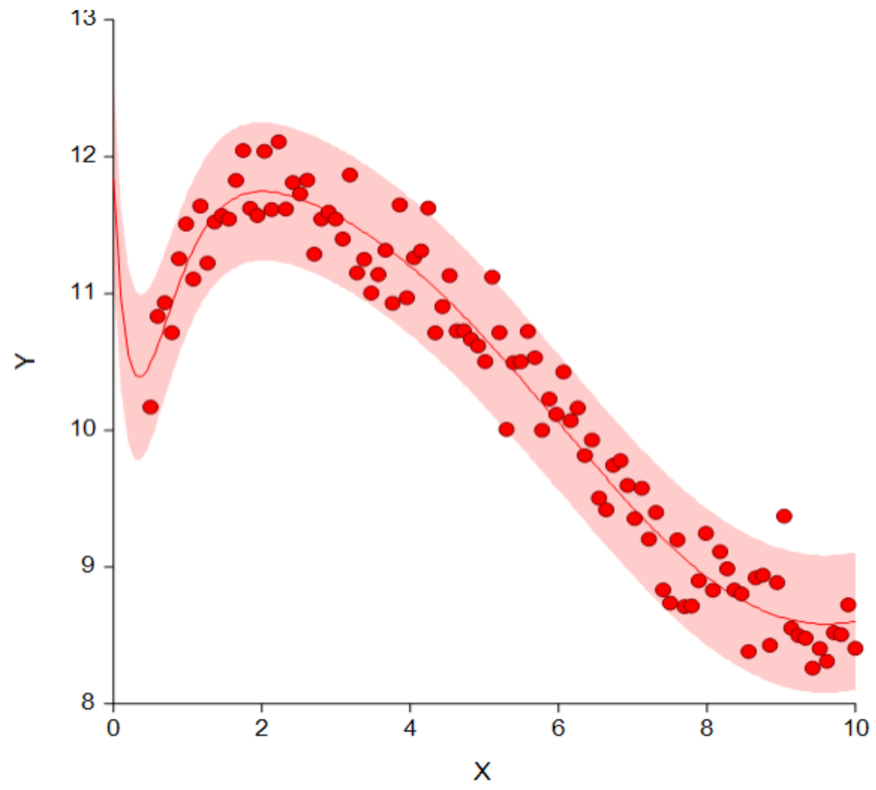
Normal Regression: Squared Loss

$D = \{(x_i, y_i)\}_{i \in I}$

$y_i = x_i + \epsilon \;\; f_\theta(x)$

Regression:

$\underset{\theta}{\operatorname{argmin}} \sum_i \| y_i - f_\theta(x_i) \|^2$

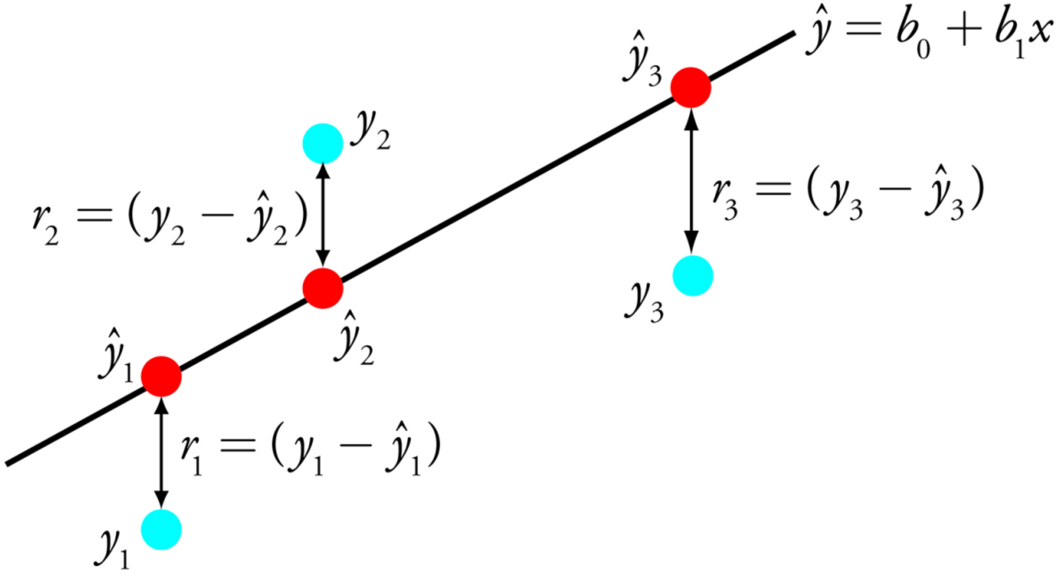


Normal Regression: Uncertainty

Regression residuals: $y_i - \hat{y}_i$

Uncertainty:

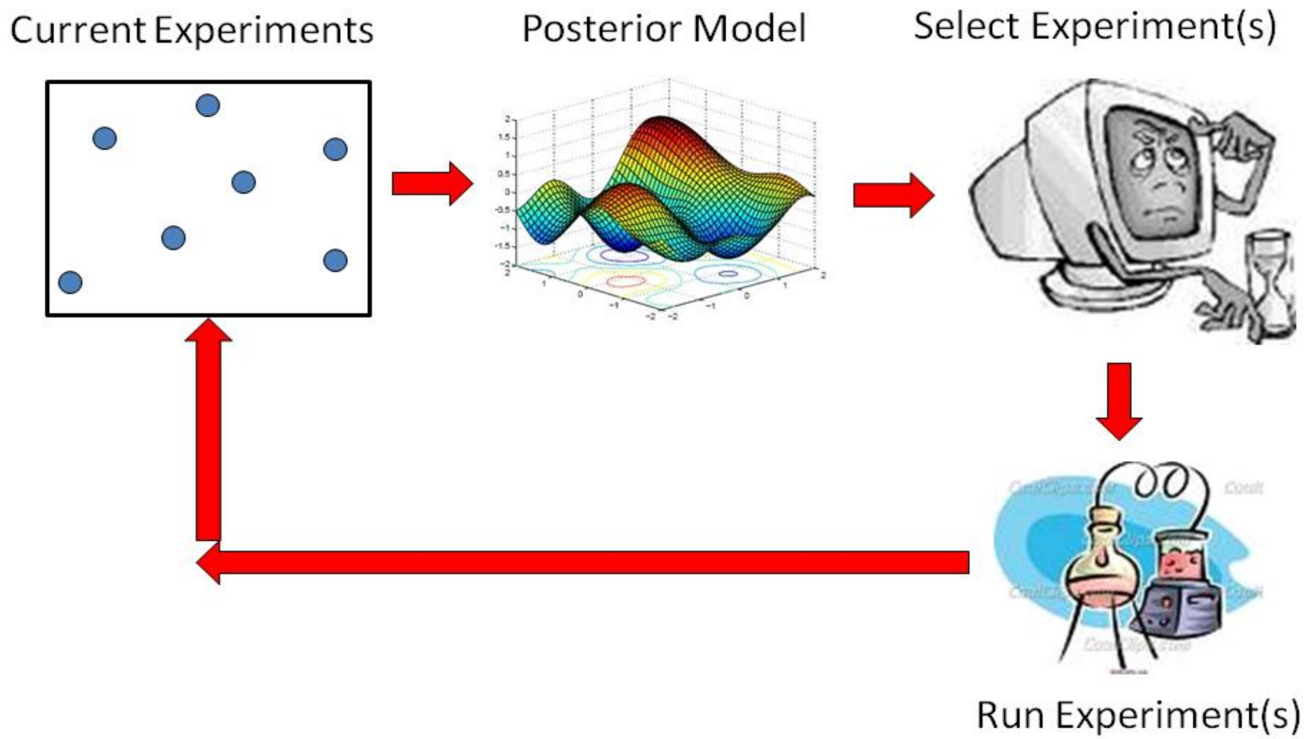
$$s_{y/x} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-2}}$$



Problems with normal regression

What is the uncertainty in parameter estimates?

Introduction: framework of Bayesian Optimization



Bayes' Rule

$$P(\theta|D) = \frac{P(D|\theta)\mathrm{P}(\theta)}{P(D)}$$

$\mathrm{P}(\theta)$ the *prior*, the distribution of the parameter(s) before any data is observed

$P(\theta|D)$ the *posterior*, the distribution of the parameter(s) after taking into account the observed data

$L(\theta|D) = P(D|\theta)$ the likelihood function, the distribution of the observed data conditional on its parameters

$P(D) = \int_\theta P(D|\theta)P(\theta)d\theta$ the marginal likelihood, the distribution of the observed data marginalized over the parameter(s)

Difference between parametric and non-parametric statistics

i.e. finite set of weights + specified model class vs general model class

Bayesian Regression

data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$

Set $P(\theta)$ and compute:

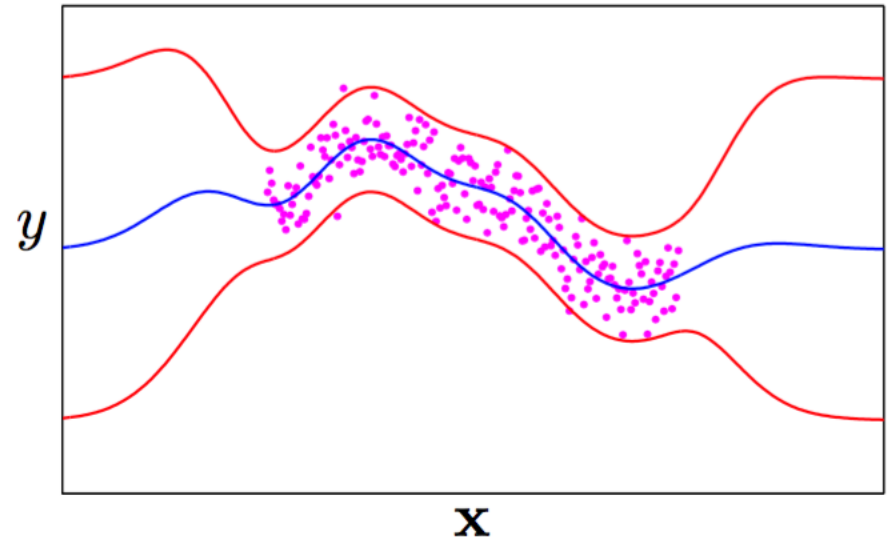
$$P(\theta|D) = \frac{P(D|\theta)\mathrm{P}(\theta)}{P(D)}$$

$$p(D|\theta) \sim \mathcal{N}(\mu, \sigma)$$

$$p(\theta) \sim \mathcal{N}(0,1)$$

$$P(D|\theta) = P(\{y_i\}|\{x_i\}, \theta)$$

$$P(D) = \int_\theta d\theta' P(D|\theta')P(\theta')$$

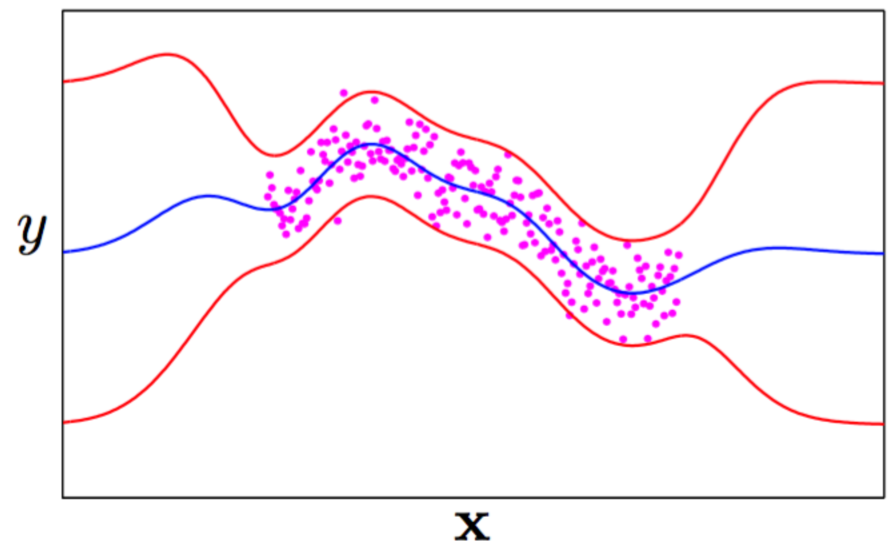


Bayesian Regression

Dataset: $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^n\} = (\mathbf{X}, \mathbf{y})$

New point (x_*, y_*)?

$$P(y_*|x_*, D) = \int P(y_*|x_*, \theta, D)P(\theta|D)d\theta$$



Gaussian Processes

Random Process Definition:

Given a probability space $(\Omega, \mathcal{F}, P)$, an $\mathbb{R}^q$-valued stochastic process is a collection of $\mathbb{R}^q$-valued random variables on Ω, indexed by a totally ordered set T. That is, a stochastic process X is a collection

$$\{X_t : t \in T\}$$

where each X_t is an $\mathbb{R}^q$-valued random variable on Ω.

Gaussian Process Definition :

A Gaussian process is a stochastic process if for every finite set of indices $t_1, \ldots, t_k$ in the index set T, $X_{t_1,t_2,\ldots,t_k} = (X_{t_1}, X_{t_2}, \ldots, X_{t_k})$ is a multivariate Gaussian random variable.

Gaussian Processes:

Alternative Gaussian Process Definition (Ghahramani):

A Gaussian process $X : \Omega \to f(\mathbb{R}^n \to \mathbb{R}^q)$ could be seen as a distribution over functions f mapping from $\mathbb{R}^n$ to $\mathbb{R}^q$, such that $(f(x_1), f(x_2), ..., f(x_k))$ is a multivariate Gaussian for every finite set of $x_1, \ldots, x_k$.

Remark

- Compared to other definition $\mathbb{R}_n$ represents index set T
- Notice that $f(x)$ are random variables

Gaussian Processes:

$f \sim \mathcal{GP}(\mu(x), K(y,z))$ denotes that f is sampled from a gaussian process and

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_k) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \dots \\ \mu(x_k) \end{bmatrix}, \begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \dots & K(x_1,x_k) \\ K(x_2,x_1) & K(x_2,x_2) & \dots & K(x_2,x_k) \\ \vdots & \vdots & & \vdots \\ K(x_k,x_1) & K(x_k,x_2) & \dots & K(x_k,x_k) \end{bmatrix} \right) \tag{1}$$

where $\mu(x)$ is called **mean function** and $K(y,z)$ is the **kernel function** !

- $\mu(x)$ could be any function!
- $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_0^+$ need to be symmetric and satisfy Mercer's condition!

Gaussian Processes: The Kernel function

- $K(x, y)$ denotes covariance between random variables $f(x)$ and $f(y)$
- $K(x, y) = 0$ implies random variable f(x) and f(y) are independent (because multivariate gaussian. . .)
- $K(x, y) = \sqrt{K(x, x)}\sqrt{K(y, y)}$ implies $f(x)$ and $f(y)$ are linearly dependent. (Proof: Determinant of Covariance Matrix vanishes)

Different Choices of Kernels make the GP emit different forms and levels of smoothness of sample functions

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

Gaussian Processes: Matern Kernels

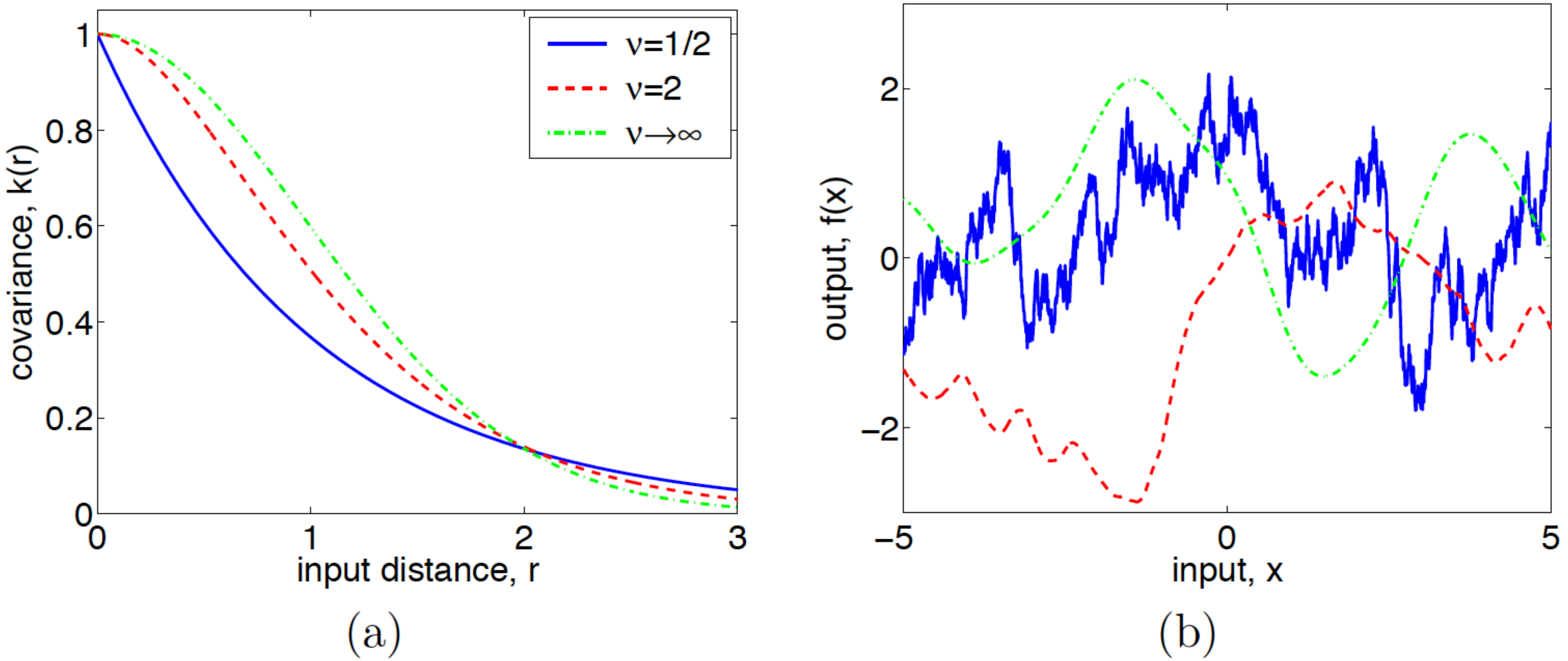


Figure 4.1: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, eq. (4.14), for different values of ν, with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

Gaussian Processes: RQ Kernels

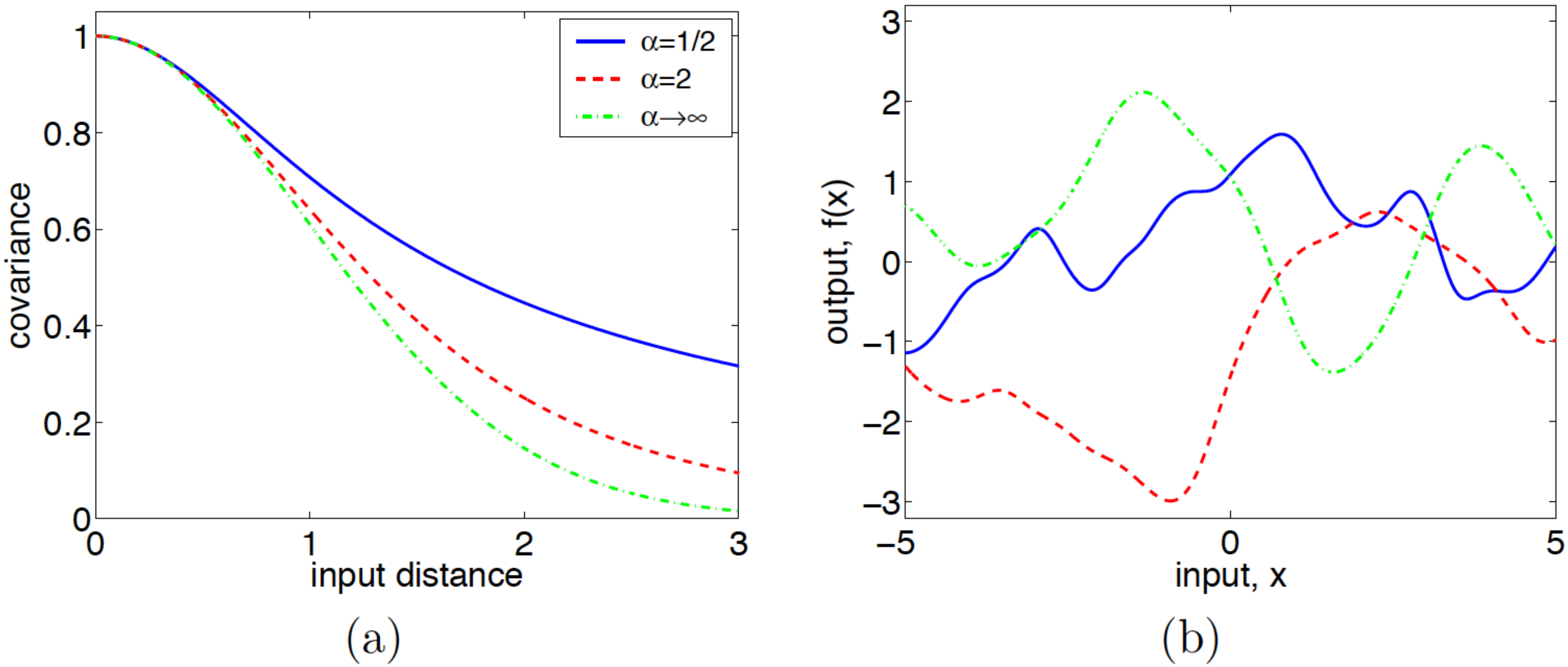
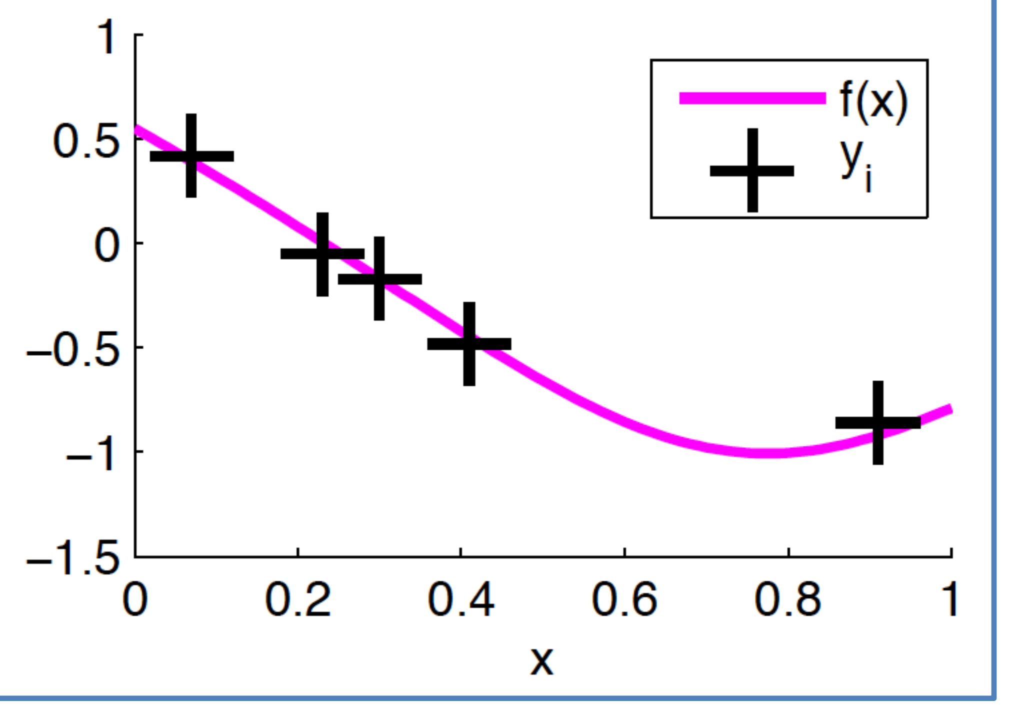


Figure 4.3: Panel (a) covariance functions, and (b) random functions drawn from Gaussian processes with rational quadratic covariance functions, eq. (4.20), for different values of α with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

Gaussian Process: Prediction problem, known kernel

Problem Statement:
Assume we know, f is sampled from $\mathcal{GP}(0, K(x, y))$, we have observed noisy measurements $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ i.i.d and we want to estimate the function f at the location x^*, i.e. we are interested in predicting $f(x^*)$.

What is p(f(x*)|{y_i}) ?



Gaussian Process: Prediction Problem Solution

Denote $\mathbf{X} = [x_1; x_2; \ldots; x_k]$ and $\mathbf{K}(X, Y)_{i,j} = K(x_i, y_j)$, then because of properties of multivariate gaussians (independence, sums of gaussians...) we yield

$$\begin{bmatrix} Y \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma \mathbf{I} & K(X, x*) \\ K(x*, X) & K(x^*, x^*) \end{bmatrix}\right)$$

Further, by using laws of conditional pdf's for multivariate Gaussians, we obtain finally the posterior distribution:

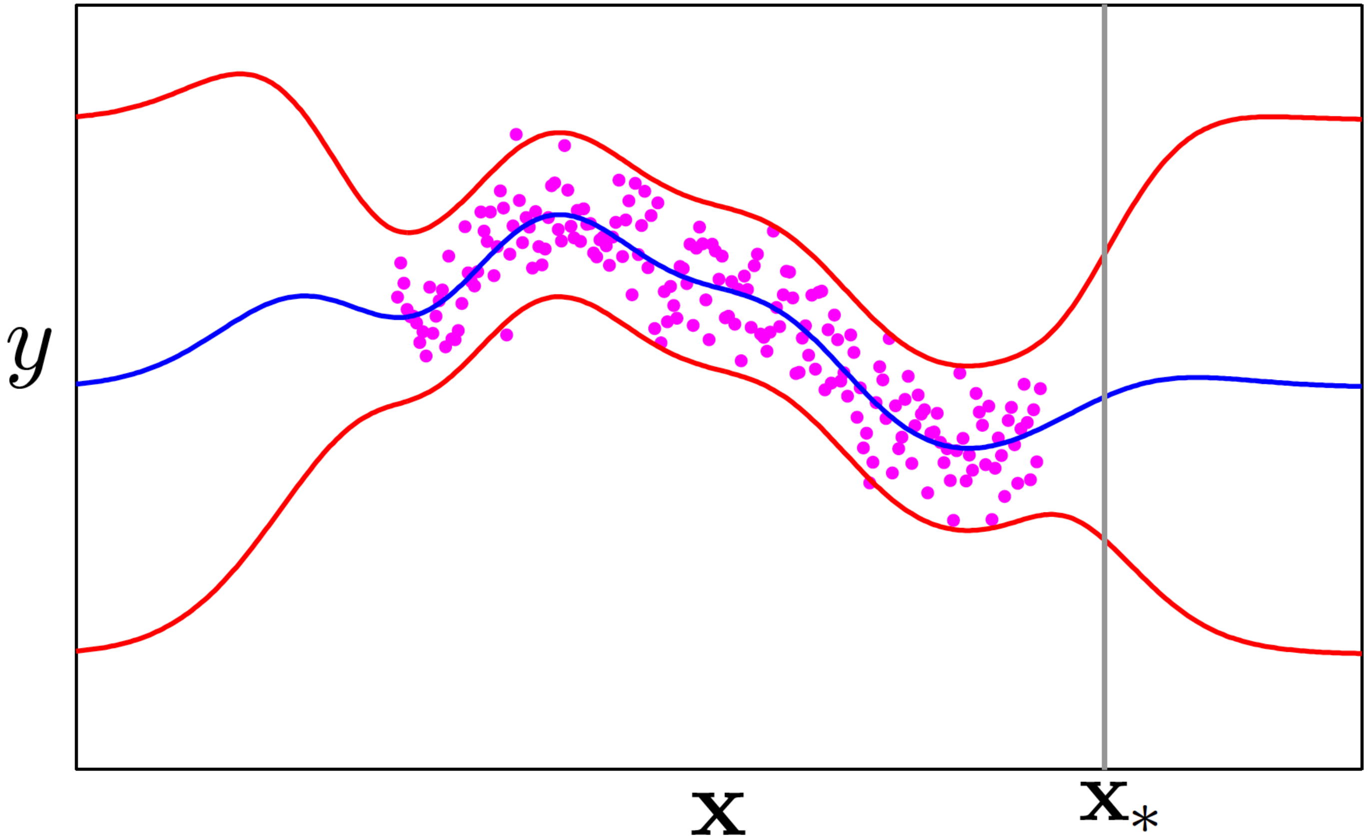
$$f(x^*) | \{y_i\} \sim \mathcal{N}\left(\bar{f}(x^*), \sigma^{2*}\right)$$

where

$$\bar{f}(x^*) = K(x^*, X)\left(K(x^*, X) + \sigma^2 \mathbf{I}\right)^{-1} Y$$
$$\sigma^{2*} = K(x^*, x^*) - K(x^*, X)\left(K(X, X) + \sigma^2 \mathbf{I}\right)^{-1} K(X, x^*)$$

Gaussian Process: Prediction Problem Solution

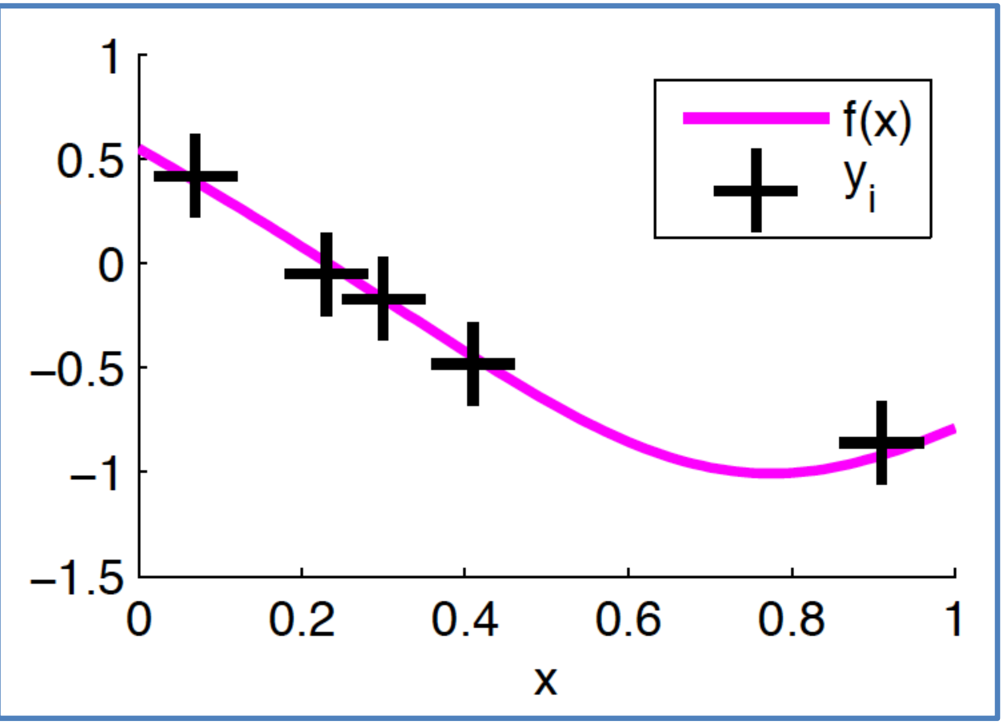
- **Minimum Variance estimator of f**
- **Uncertainty envelop certifying quality of predictions**



Gaussian Process: Prediction problem, unknown kernel

Problem Statement:
Assume we know, f is sampled from $\mathcal{GP}(0, K(x, y, \theta))$, we have observed noisy measurements $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ i.i.d and we want to estimate the function f at the location x^*, i.e. we are interested in predicting $f(x^*)$. Also θ is an unknown parameter on which the Kernel function depends.

What is p(f(x*)|{y_i}) ?



Gaussian Process: Unknown Kernel Parameter

Approach 1: Use point-estimate of theta

(Recall notation: $\mathbf{X} = [x_1; x_2; \ldots; x_k]$, $\mathbf{Y} = [y_1; y_2; \ldots; y_k]$)

Step 1: Estimate parameter from observations

ML approach

- We can compute $p_X(Y|\theta)$ since $Y|\theta \sim \mathcal{N}(0, K(X, X, \theta))$
- $\hat{\theta}_{ML} = \text{argmax}_\theta p_X(Y|\theta)$.

MAP approach

- posterior $p_X(\theta|Y) = \frac{p_X(Y|\theta)p(\theta)}{\int p_X(Y|\theta)p(\theta)d\theta}$
- $\int p_X(Y|\theta)p(\theta)d\theta$ only function of Y.
- $\hat{\theta}_{MAP} = \text{argmax}_\theta p_X(Y|\theta)p(\theta)$.

Step 2: Compute p(f(x*)|{y_i},theta) as before

Gaussian Process: Unknown Kernel Parameter

(Approach 2: Marginalize the parameter out)

(Recall notation: $\mathbf{X} = [x_1; x_2; \ldots; x_k]$, $\mathbf{Y} = [y_1; y_2; \ldots; y_k]$)

$$p_X(f(x*)|Y) = \int p_X(f(x*)|\theta, Y)p(\theta|Y)d\theta$$

$$= \int p_X(f(x*)|\theta, Y)\left(\frac{p_X(Y|\theta)p(\theta)}{\int p_X(Y|\theta)p(\theta)d\theta}\right)d\theta$$

Gaussian Process: Kernel Parameter

Approach 1 (Point-estimation)

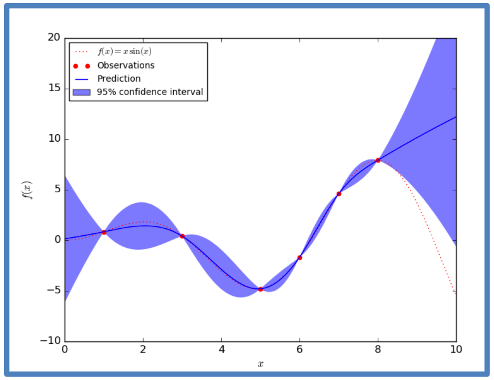
- Easier, gives direct analytical solutions, everything remains Gaussian
- Can have problem with overfitting

Approach 2 (Marginalizing out the parameter)

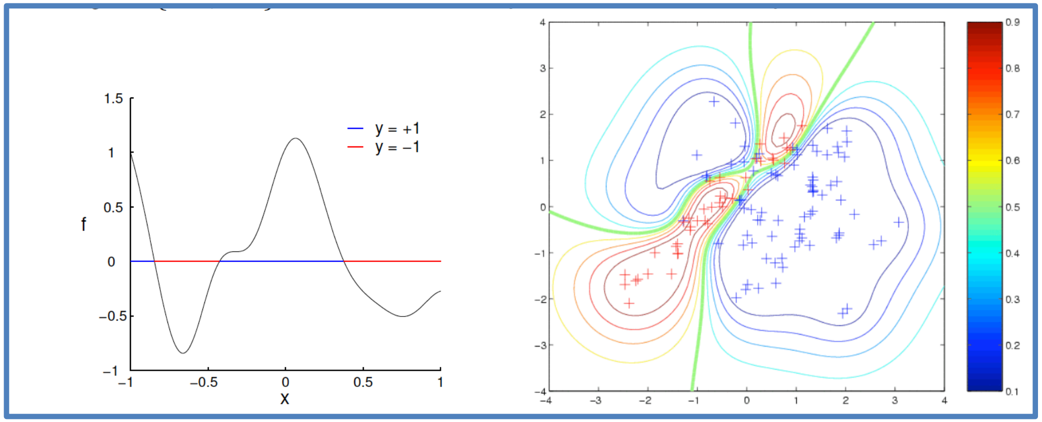
- Difficult integration, analytical sol. only if you assume "right" prior for theta
- Better generalization, accounting for uncertainty in theta

Gaussian Processes: Applications

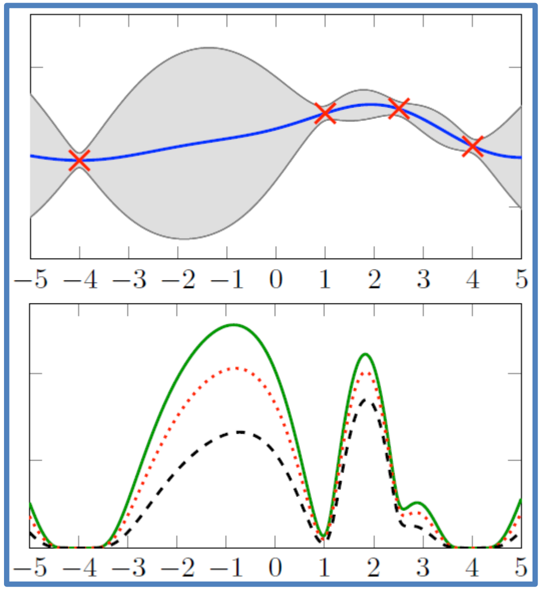
Nonlinear Regression



Classification



!!Bayesian Optimization!!



Bayesian Optimization

General Purpose

Online Optimization of f when f is not a priori known and evaluating f is expensive. Naturally handles uncertainty.

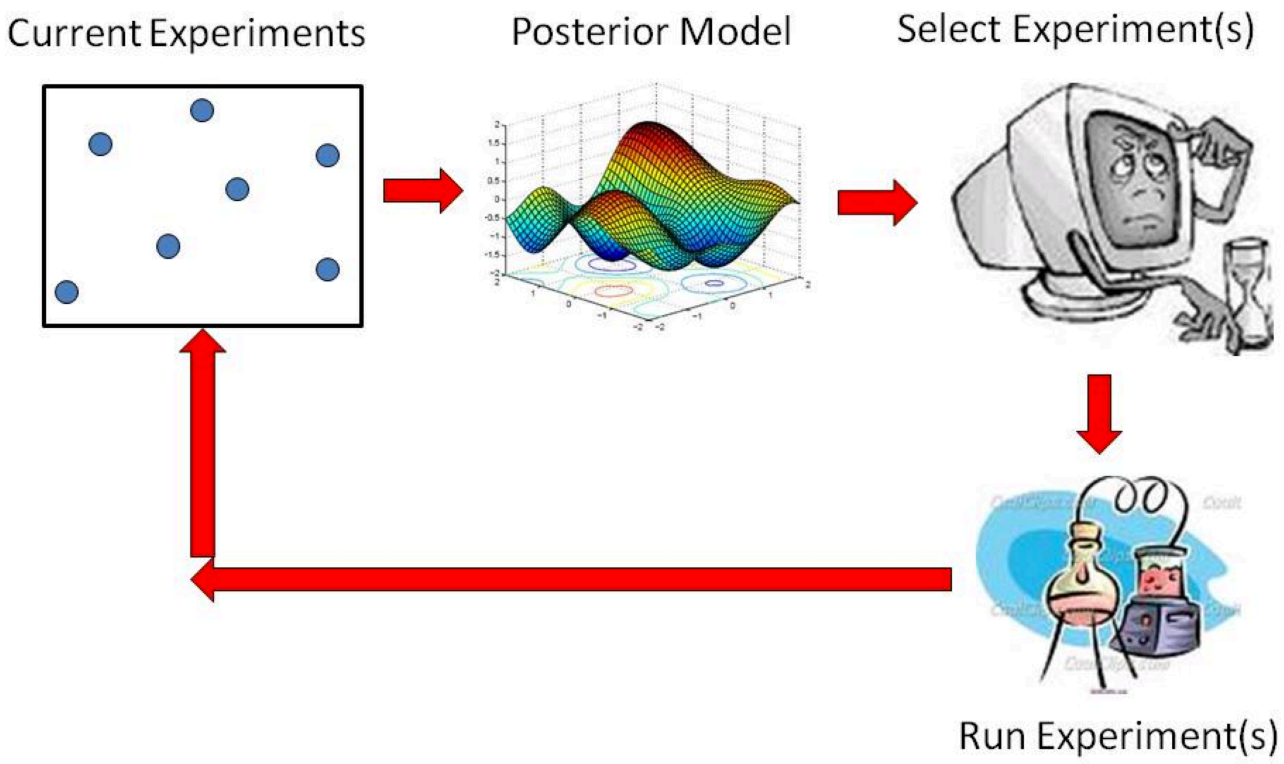
$$\max_{x \in \mathcal{X}} f(x)$$

Example Application:

(Today's Paper)
Tuning of Hyperparameters of ML algorithms

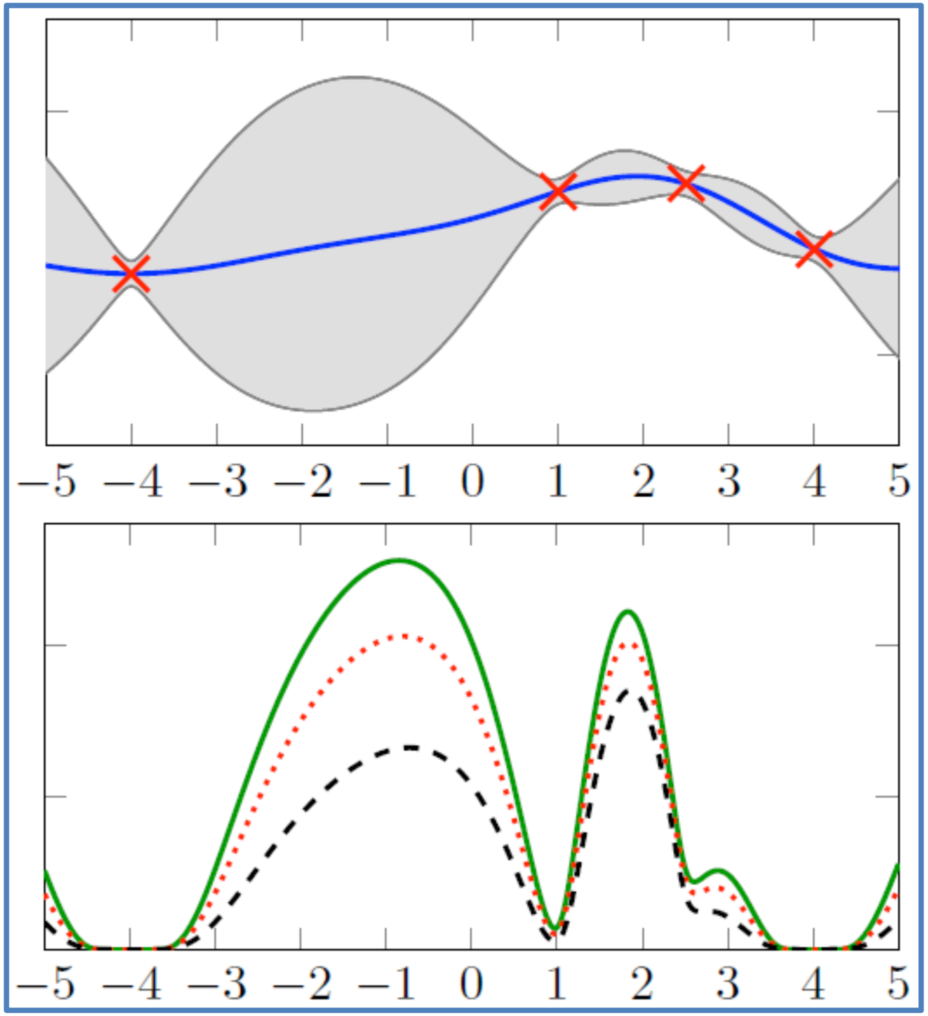
Bayesian Optimization

General Idea:



Source: http://javad-azimi.com/

Bayesian Optimization



General procedure:

1. Estimate your current posterior belief
   about the function using GP
   and observations
2. Use this belief and a strategy to hypothesize about minimizer x*
   1. Encoded via maximization of acquisition function
3. Request sample y*=f(x*) and go to 1.

Source: http://mlg.eng.cam.ac.uk/amar/pics/TPbayesopt.png

Bayesian Optimization

Assume, we have:

- noiseless observations $y_i = f(x_i)$
- f is sampled from a gaussian process $f \sim \mathcal{GP}(\mu(x,\theta), K(x,y,\theta)$
- Do not know θ but assume some prior $p(\theta)$.

Objective: Find $\max_{x \in \mathcal{X}} f(x)$ and $x^* = \mathrm{argmax}_{x \in \mathcal{X}} f(x)$:

General Procedure

for k=1,2,...

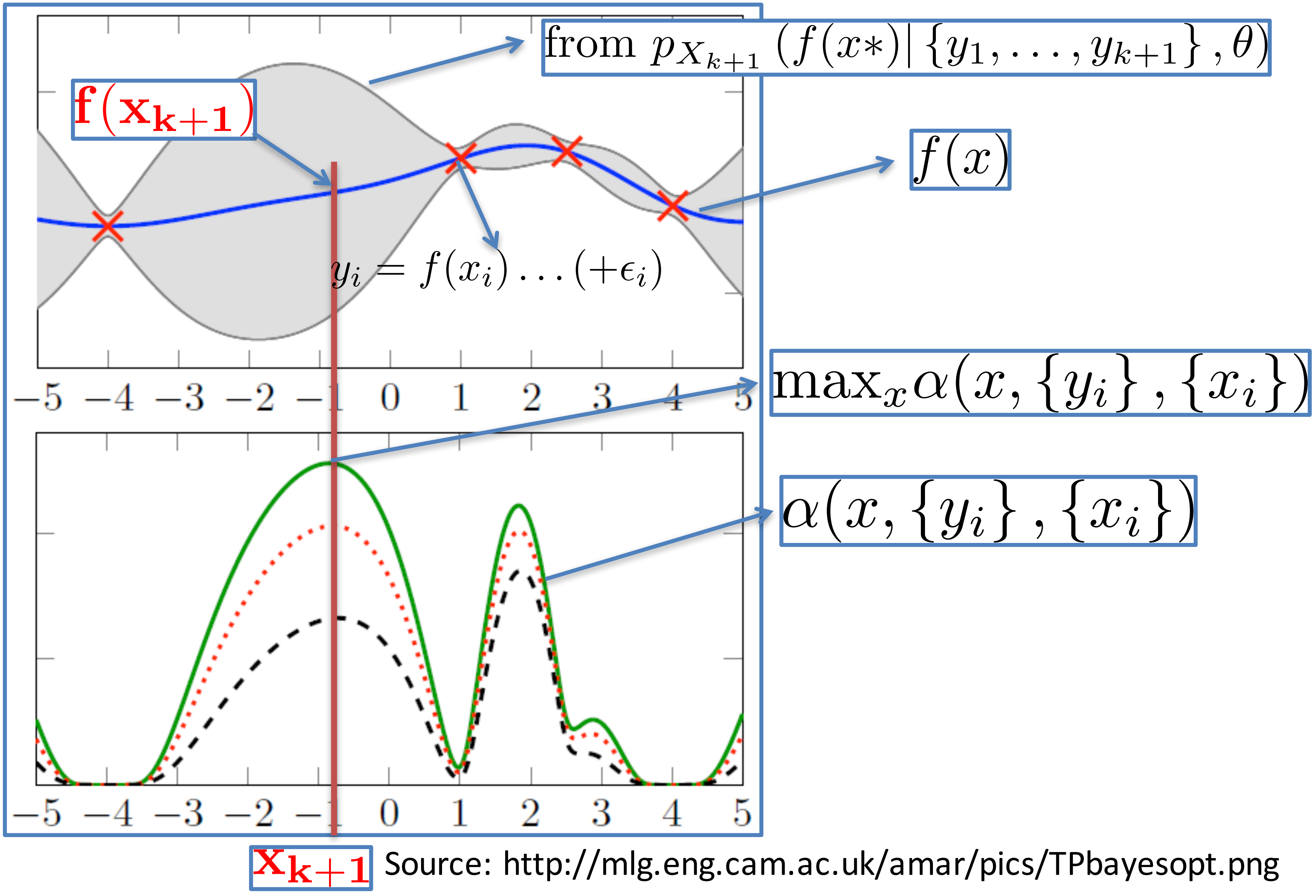
1. Select next sample point/ index x_{k+1} based on maximizing a **acquisition function** α:

$$x_{k+1} = \mathrm{argmax}_x \alpha(x, \{y_i\}, \{x_i\}, \tau_{k+1}) \tag{1}$$

2. query objective function to obtain $y_{k+1} = f(x_{k+1})$
3. augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{k+1}, y_{k+1})\}$ and hyperparameter τ_k
4. update statistical model of function/ posterior:
   (a) (Estimate $\hat{\theta}$ through ML or MAP)
   (b) Compute new process posterior $p_{X_{k+1}}(f(x*)| \{y_1, \ldots, y_{k+1}\}, \theta)$
   (or just $p_{X_{k+1}}(f(x*)| \{y_1, \ldots, y_{k+1}\}, \hat{\theta}))$

end for

Bayesian Optimization



Source: http://mlg.eng.cam.ac.uk/amar/pics/TPbayesopt.png

Bayesian Optimization

Acquisition fcn have often general form of

1. If marginalizing the parameter theta out:

$$\alpha(x, \{y_i\}, \{x_i\}, \tau) = \mathbb{E}_{\theta|\{y_i\}}\mathbb{E}_{f(x)|\theta,\{y_i\}}\left[U(x, f(x), \theta, \tau)\right]$$

$\sim p_{\{x_i\}}(f(x)|\{y_i\}, \theta)$
is posterior derived from GP assumption!

2. If making point-estimate of theta:

$$\alpha(x, \{y_i\}, \{x_i\}, \tau) = \mathbb{E}_{f(x)|\hat{\theta},\{y_i\}}\left[U(x, f(x), \hat{\theta}, \tau)\right]$$

Bayesian Optimization

- Utility function give score, how good x is as a minimizer candidate
- Acquisition functions can be designed wrt. Exploration/exploitation trade-off
- Generalization of Multi-Arm Bandit Problem to the general continuous case

Bayesian Optimization Acquisition Functions

Probability of Improvement

With $U(x, y, \theta, \tau) = \mathbb{I}(y > \tau)$ we get

$$\alpha_{PI}(x, \{y_i\}, \{x_i\}, \tau) = \mathbb{P}\left(f(x) > \tau \middle| \{y_i\}, \{x_i\}\right) \ldots (= \Phi(\frac{\mu_{f(x)|Y,\theta} - \tau}{\sigma_{f(x)|Y,\theta}(x)}))$$

Tau is usually picked as the biggest f(x_i) so far!

Bayesian Optimization Acquisition Functions

Expected Improvement

With $U(x, y, \theta, \tau) = (y - \tau)\mathbb{I}(y > \tau)$ we get

$$\alpha_{EI}(x, \{y_i\}, \{x_i\}, \tau) = \mathbb{E}\left(f(x) - \tau | \{y_i\}, \{x_i\}, f(x) >= \tau\right)\mathbb{P}\left(f(x) >= \tau\right)$$

$$\ldots (= \left(\mu_{f(x)|Y,\theta} - \tau\right)\Phi(\frac{\mu_{f(x)|Y,\theta} - \tau}{\sigma_{f(x)|Y,\theta}(x)}) + \sigma_{f(x)|Y,\theta}(x)\phi(\frac{\mu_{f(x)|Y,\theta} - \tau}{\sigma_{f(x)|Y,\theta}(x)}))$$

Tau is usually picked as the biggest f(x_i) so far!

Bayesian Optimization Acquisition Functions

GP Upper Confidence Bound

Either point-estimated

$$\alpha_{UCB}(x, \{y_i\}, \{x_i\}, \beta) = \mu_{f(x)|Y,\hat{\theta}} + \beta\sigma_{f(x)|Y,\hat{\theta}}(x)$$

or marginalized

$$\alpha_{UCB}(x, \{y_i\}, \{x_i\}, \beta) = \int \left(\mu_{f(x)|Y,\theta} + \beta\sigma_{f(x)|Y,\theta}(x)\right)\left(\frac{p_X(Y|\theta)p(\theta)}{\int p_X(Y|\theta)p(\theta)d\theta}\right)d\theta$$

Beta Parameter trades off exploration vs. exploitation like in MAB

Bayesian Optimization Acquisition Functions

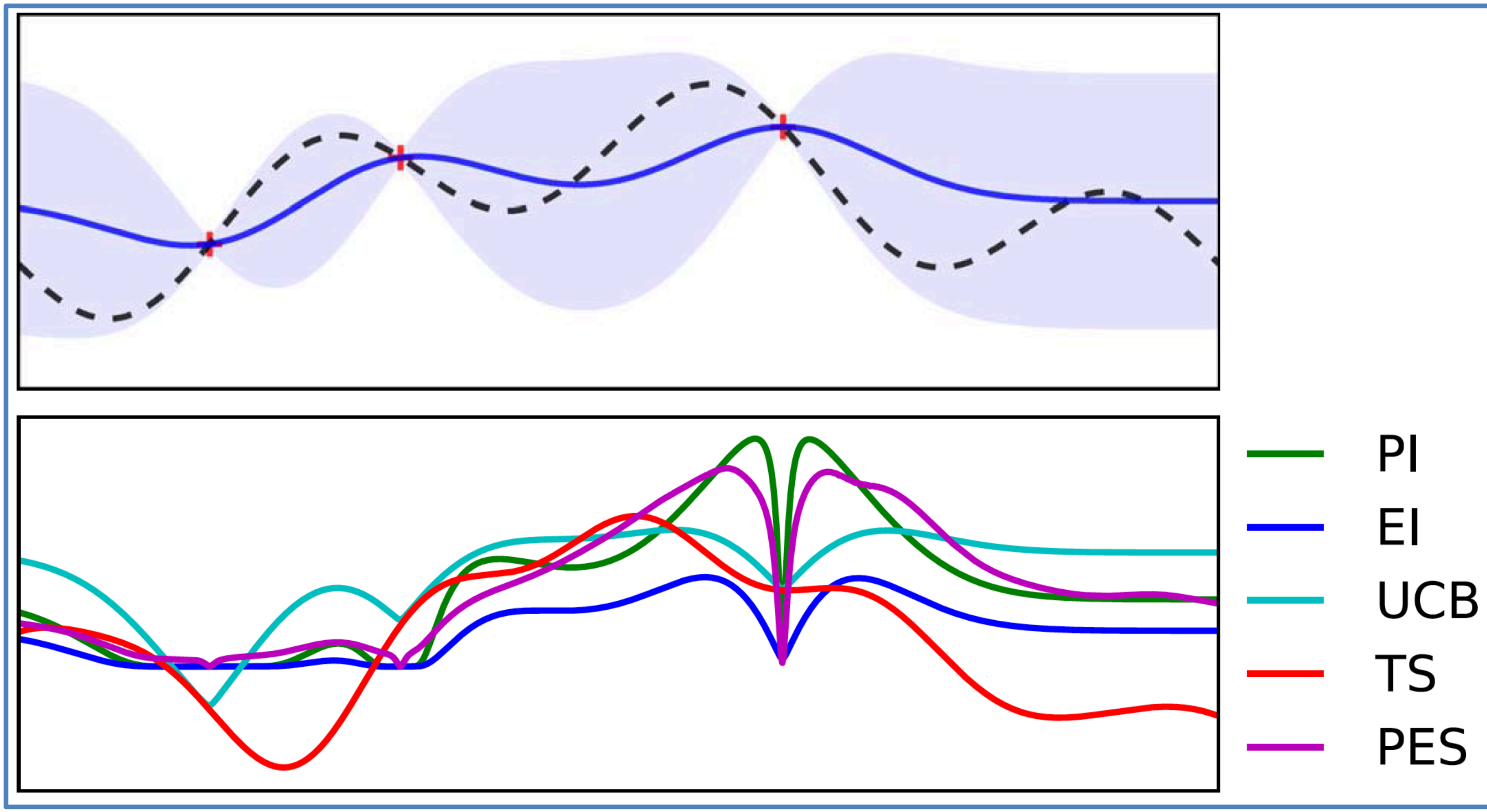
Thompson Sample

$$\alpha_{TS}(x, \{y_i\}, \{x_i\}) = f_{\{y_i\}}$$

where

$$f_{\{y_i\}} \sim \mathcal{GP}\left(\mu(x, \hat{\theta}), K(x, y, \hat{\theta}) \middle| \{y_i\}, \{x_i\}\right)$$

Bayesian Optimization: Comparison of Acquisition Functions



Bayesian Optimization Dealing with theta

1. Point-estimate of θ via ML or MAP:
   - easy and tractable to compute α, but can cause overfitting
2. Marginalizing θ "out of the α function"
   - hard to do due to integration, but gives better generalization.
   - Solution: Quadrature Approximation
   - Solution: Monte Carlo techniques (SMC, MCMC), which try sampling $\{\theta_i\}_{i=1..M} \sim p(\theta, Y)$ and approximation $\mathbb{E}_{\theta|Y}\left(F(\theta)\right) \approx \frac{1}{M}\sum_{i=1}^{M} F(\theta_i)$

Bayesian Optimization by

Snoek, Larochelle, Adams

1. Assume Gaussian prior of θ

for k=1,2,...

1. Select next sample point/ index x_{k+1} based on maximizing a **acquisition function** α:

$$x_{k+1} = \text{argmax}_x \alpha(x, y_i, x_i) \tag{1}$$

2. query objective function to obtain $y_{k+1} = f(x_{k+1})$
3. augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{k+1}, y_{k+1})\}$
4. update statistical model of function/ posterior:
   (a) (Estimate θ)
   (b) Gaussian process posterior $p_{x_i}\left(f(x*) | \{y_i\}, \theta\right)$

end for

Bayesian Optimization by

Snoek, Larochelle, Adams

2. Choice of Kernel for GP:

$$K_{\mathsf{M52}}(\mathbf{x},\mathbf{x}') = \theta_0 \left(1 + \sqrt{5r^2(\mathbf{x},\mathbf{x}')} + \frac{5}{3}r^2(\mathbf{x},\mathbf{x}')\right) \exp\left\{-\sqrt{5r^2(\mathbf{x},\mathbf{x}')}\right\}$$

for k=1,2,...

1. Select next sample point/ index x_{k+1} based on maximizing a **acquisition function** α:

$$x_{k+1} = \text{argmax}_x \alpha(x, y_i, x_i) \tag{1}$$

2. query objective function to obtain $y_{k+1} = f(x_{k+1})$
3. augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{k+1}, y_{k+1})\}$
4. update statistical model of function/ posterior:
   (a) (Estimate θ)
   (b) Gaussian process posterior $p_{x_i}\left(f(x*) | \{y_i\}, \theta\right)$

end for

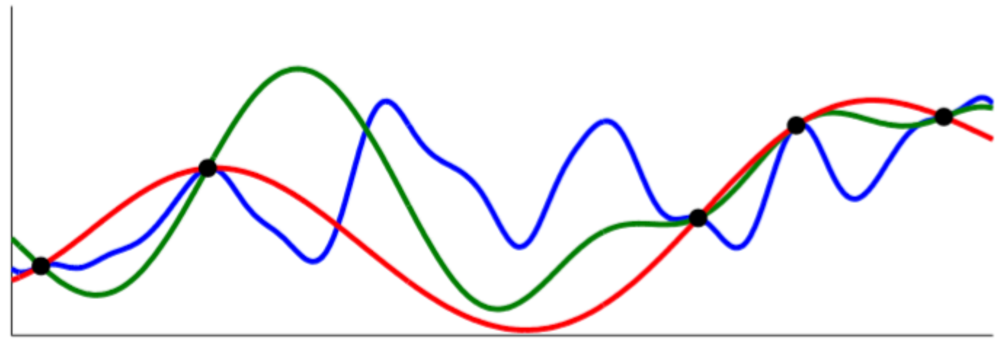
Bayesian Optimization by

Snoek, Larochelle, Adams

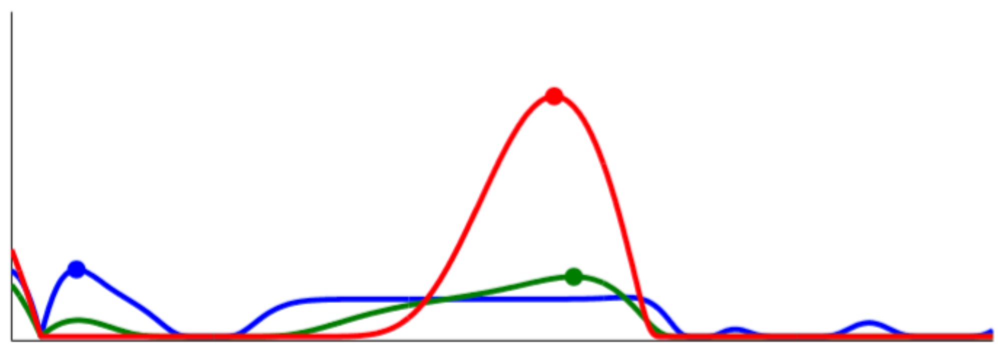
3. Choice of Acquisition function for GP:
   1. Expected Improvement per second
   2. Marginalizing out θ

$$a_{\mathsf{EI}}(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta) = \sigma(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta)\,(\gamma(\mathbf{x})\,\Phi(\gamma(\mathbf{x})) + \mathcal{N}(\gamma(\mathbf{x})\,;\,0, 1))$$

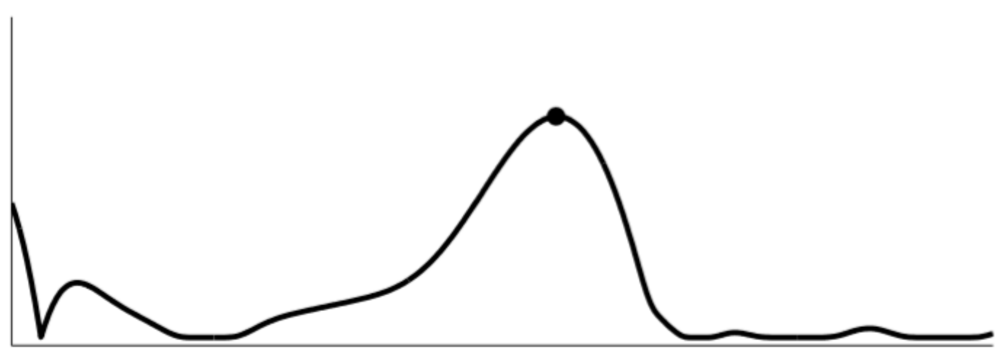
$$\hat{a}(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}) = \int a(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta)\,p(\theta \,|\, \{\mathbf{x}_n, y_n\}_{n=1}^{N})\,\mathrm{d}\theta,$$



(a) Posterior samples under varying hyperparameters



(b) Expected improvement under varying hyperparameters



(c) Integrated expected improvement

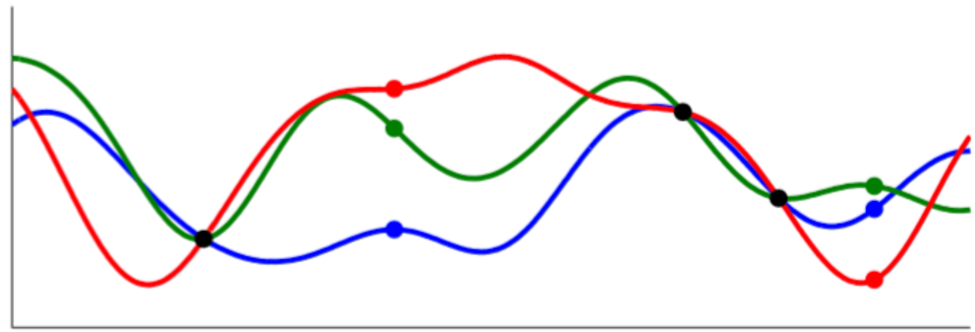
Bayesian Optimization by

Snoek, Larochelle, Adams

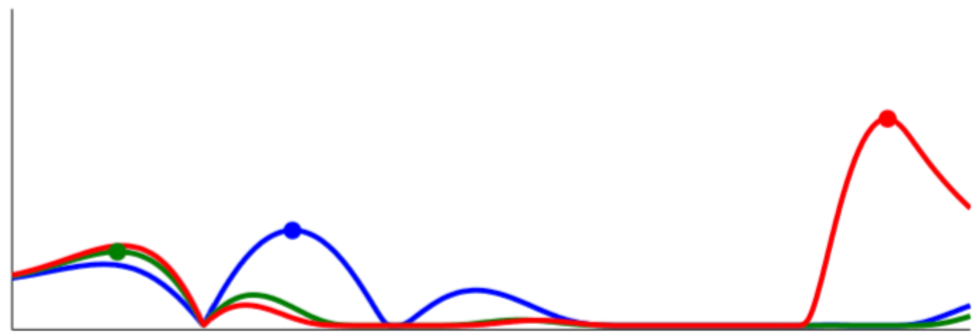
4. Computation:

1. Monte Carlo for parallelization and computation of alpha

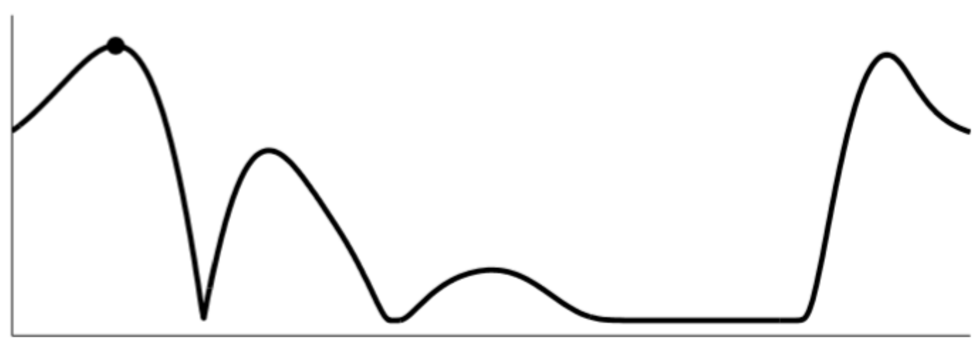
$$\hat{a}(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta, \{\mathbf{x}_j\}) = \int_{\mathbb{R}^J} a(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta, \{\mathbf{x}_j, y_j\})\, p(\{y_j\}_{j=1}^J \,|\, \{\mathbf{x}_j\}_{j=1}^J, \{\mathbf{x}_n, y_n\}_{n=1}^N)\, \mathrm{d}y_1 \cdots \mathrm{d}y_J.$$



(a) Posterior samples after three data



(b) Expected improvement under three fantasies

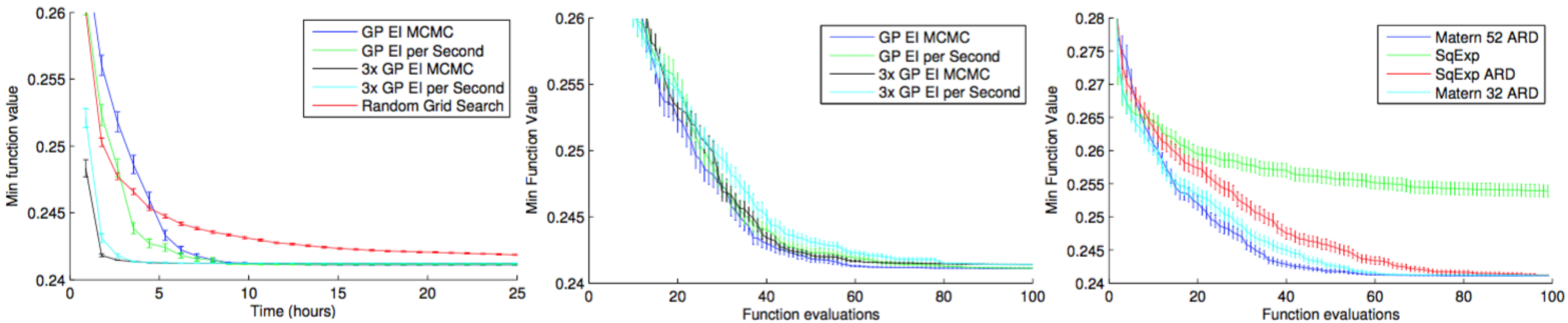


(c) Expected improvement across fantasies

Bayesian Optimization by

Snoek, Larochelle, Adams

5. Comparison on 3 ML algorithms Hyper-parameter tuning



(a) (b) (c)