Bayseian Optimization

2019

Gaussian Process

- Is a Gaussian distribution over functions.

- f is a Gaussian process if for every finite set of points t_1, ..., t_n: $(f(t_1), ..., f(t_n))$ is a multivariate Gaussian random variable.

- We need mean and variance at every point and covariance between every pair of point to identify a Gaussian process.

- If we assume mean is zero we only need to find variances and covariances to determine the process.

Gaussian Process cont'd

- For a set of points like t_1, ..., t_n the covariance matrix can be constructed by finding covariances of every pair of points and variances of individual points.

- For covariance we use a function called the kernel function, which takes two points t_1, t_2 and its output is $cov[f(t_1), f(t_2)]$. This function computes variance at each point as well.

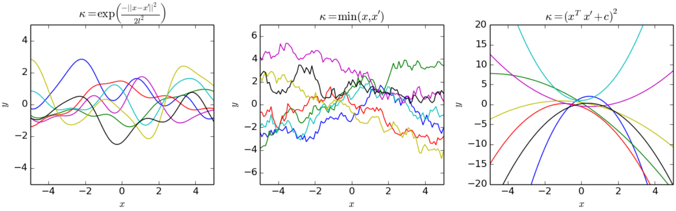


Image taken from wikipedia.

Gaussian Distribution Recap

- For identifying a multivariate Gaussian, we only need to know its mean and covariance matrix.
- For a multivariate Gaussian distribution, both the marginal and conditional distributions are Gaussian and the parameters(mean and covariance) can be computed analytically.

Marginal of a Gaussian

If $x = (x_1, x_2)$ is guassian with parameters:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} = \Sigma_{12}^T & \Sigma_{22} \end{pmatrix}$$

Then the marginals are:

$$p(x_1) = \mathcal{N}(x_1 | \mu_1, \Sigma_{11}), p(x_2) = \mathcal{N}(x_2 | \mu_2, \Sigma_{22})$$

Conditional of a Gaussian

- The conditional probability(same setting as the previous slide):

$$p(x_1|x_2) = \mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2})$$

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

- The point is all parameters can be computed analytically.

Search for Good Hyperparameters

- We have an objective function, we care about generalization performance. Use cross validation to measure parameter quality.
- We can evaluate the objective pointwise, but do not have an easy functional form or gradients.
- How do people currently search?
  - Grid search
  - Random search
- Learning the parameters and computing the validation loss is computationally costly.

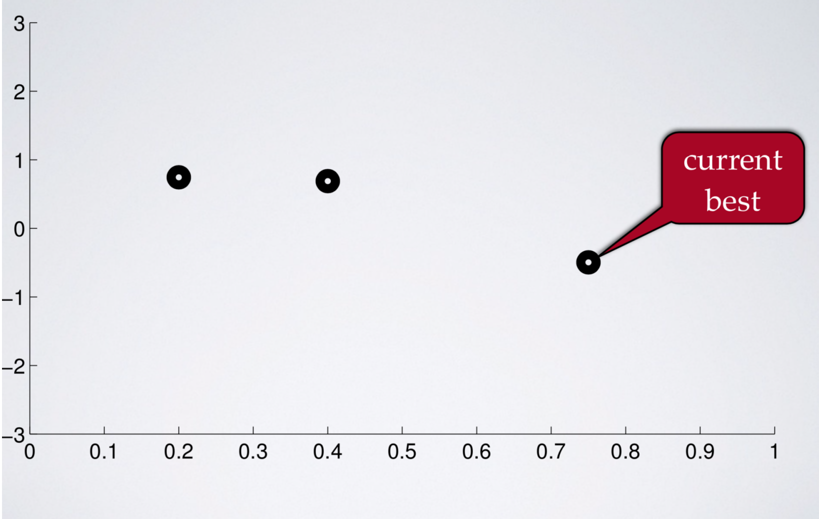
Bayesian Optimization

- Build a probabilistic model for the objective using Gaussian process.
- Compute the posterior predictive distribution. Since our model is a Gaussian process this can be computed analytically.
- Optimize a cheap proxy function instead. Make the proxy function exploit uncertainty to balance exploration against exploitation.(More on this later)

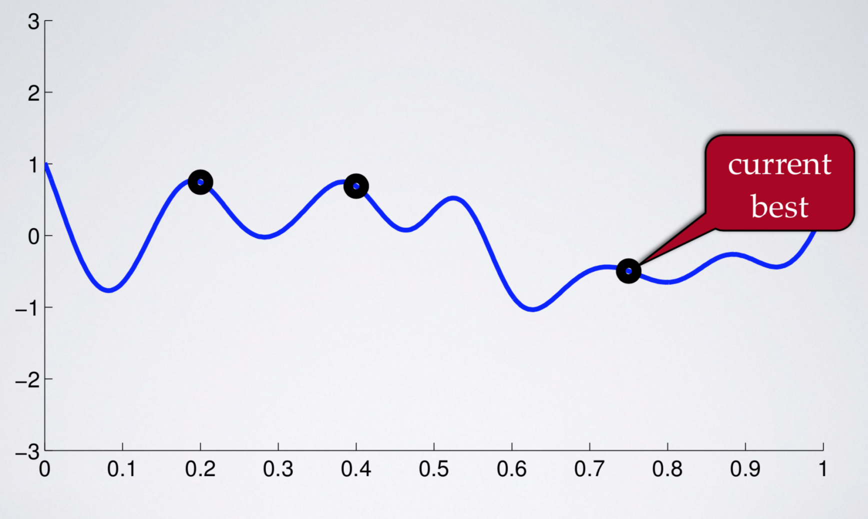
Bayesian Optimization

- Fix a Kernel function.
- Evaluate the objective at a random point first.
- Now using the kernel function and functions value at random point we can compute the posterior predictive mean and variance for every point. Since marginal distribution of Gaussian is Gaussian, we have a distribution for objective function values at every point.
- Now using these distributions and an acquisition function(proxy function) to decide which point to evaluate next. Repeat!
- Keep track of the minimum.

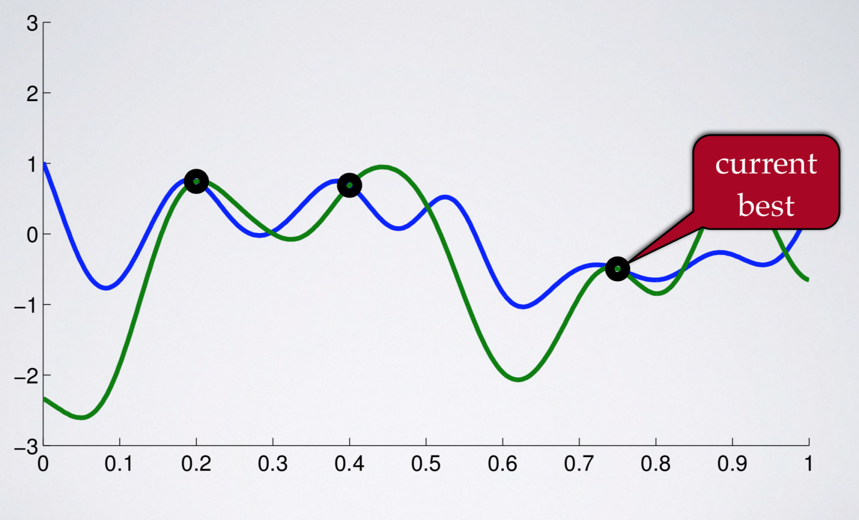
How to decide for the next point



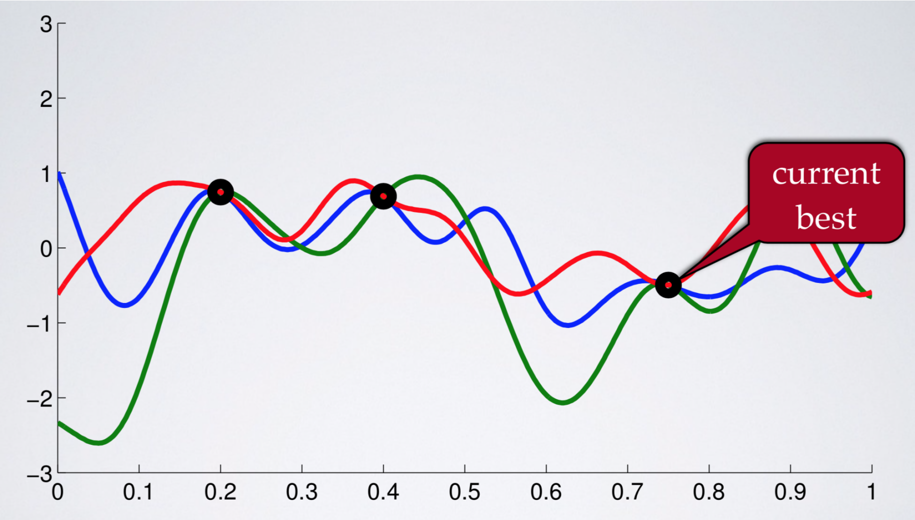
How to decide for the next point



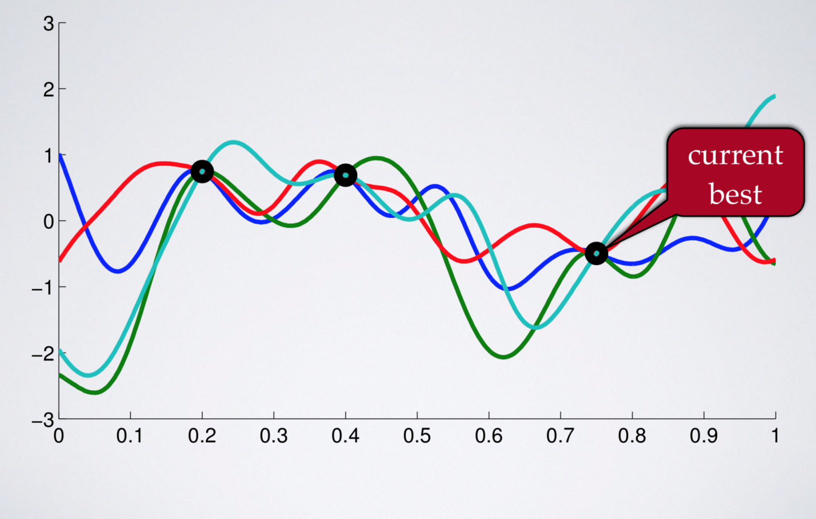
How to decide for the next point



How to decide for the next point

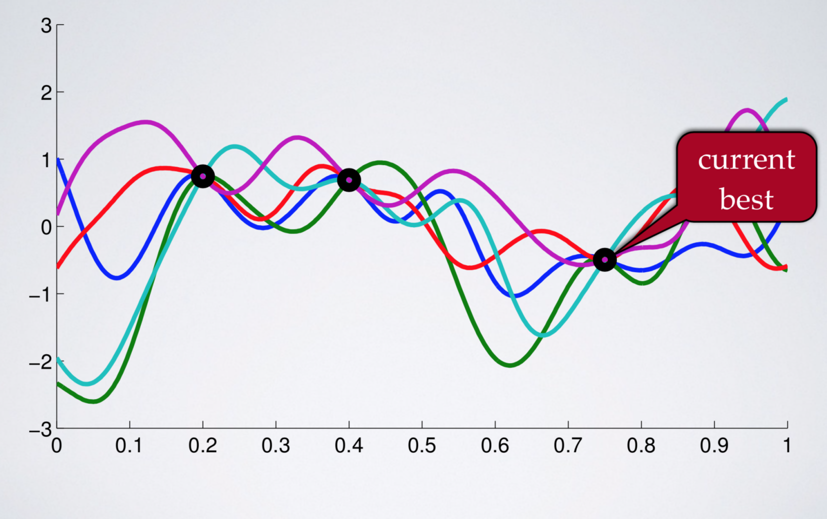


How to decide for the next point



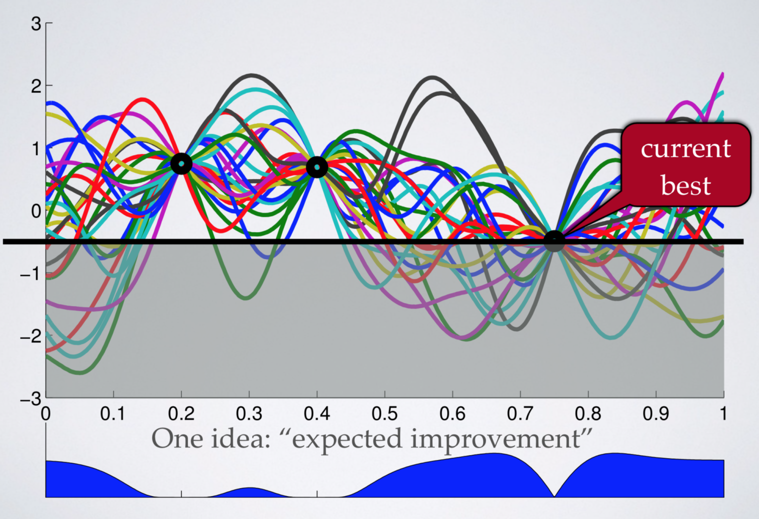
current best

How to decide for the next point



current best

How to decide for the next point



One idea: "expected improvement"

Images taken from Ryan P. Adams's Slides.

Acquisition Function

- After computing posterior predictive distribution over points, we need a criteria for choosing the next point to evaluate.

- Acquisition function is an easy to evaluate function based on the distribution over points(which can be computed based on mean and covariances since distributions are Gaussian) which helps us choose the next point.

- We should use a function that balances exploration and exploitation
  - Exploration: Seek places with high variance.
  - Exploitation: Seek places with low mean.

Expected Improvement

- Idea: For a given point what is the expected value of improvement, while neglecting the cases when the new value is worse than previous minimum.
- If we did worse we will keep the previous min so there is not much to lose, so concentrate on gain.
- $EI(x) = \mathbb{E}[max(0, f^{min} - f_{n+1}(x) - \xi)|\mathcal{D}_n]$
- Next point should maximize expected improvement.
- ξ Controls exploration and exploitation balance.
- EI can be computed analytically:

$$EI(x) = \begin{cases} (f^{min} - \mu(x) - \xi)\Phi(Z) + \sigma(x)\phi(Z) & \sigma(x) > 0 \\ 0 & \sigma(x) = 0 \end{cases}$$

$$Z = \frac{f^{min} - \mu(x) - \xi}{\sigma(x)}$$

Probability of Improvement

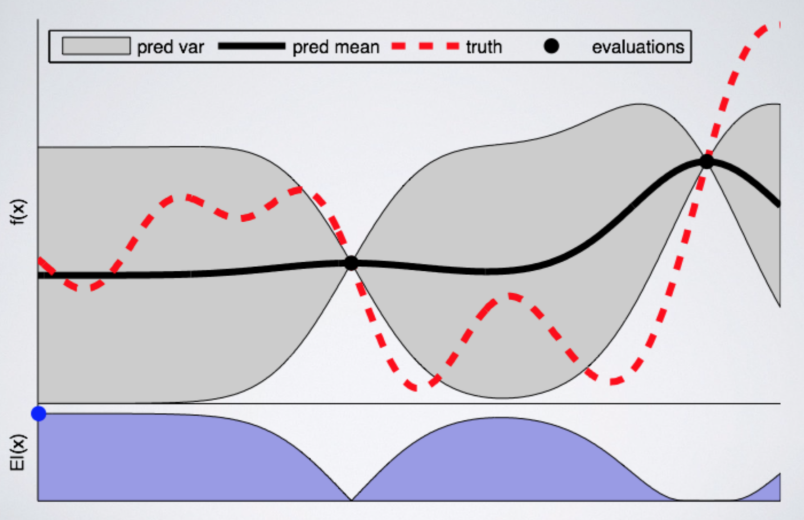
- Idea: For a given point, what is the probability of getting a better result.
- $PI(x) = \mathbb{P}(f_{n+1}(x) \leq f^{min} - \xi | \mathcal{D}_n)$
- Next point should maximize probability of improvement.
- ξ Controls exploration and exploitation balance.
- PI can be computed analytically:

$$PI(x) = \Phi(\frac{f^{min} - \mu(x) - \xi}{\sigma(x)})$$

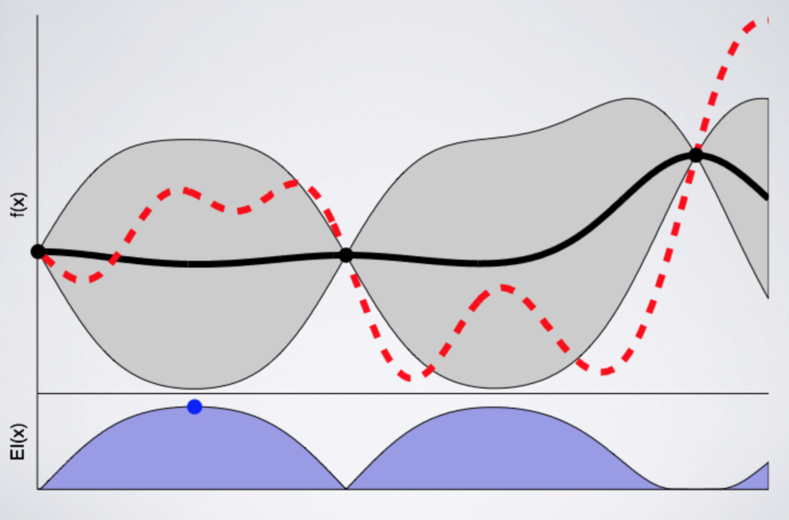
Lower(Upper) Confidence Bound

- Idea: Compute a confidence interval(For example %95), and choose the point with lowest bound.
- $LCB(x) = \mu(x) - \kappa\sigma(x)$
- Next point should minimize(maximize) Lower(Upper) Confidence.
- κ Controls exploration and exploitation balance.

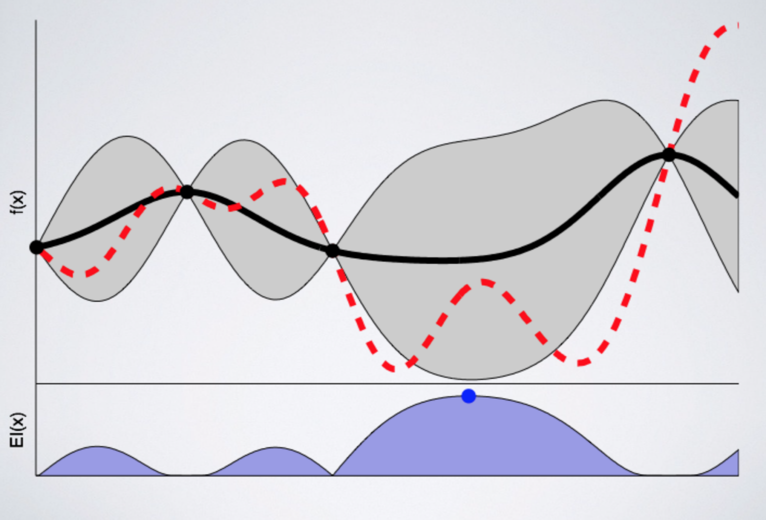
Illustrating Bayesian Optimization



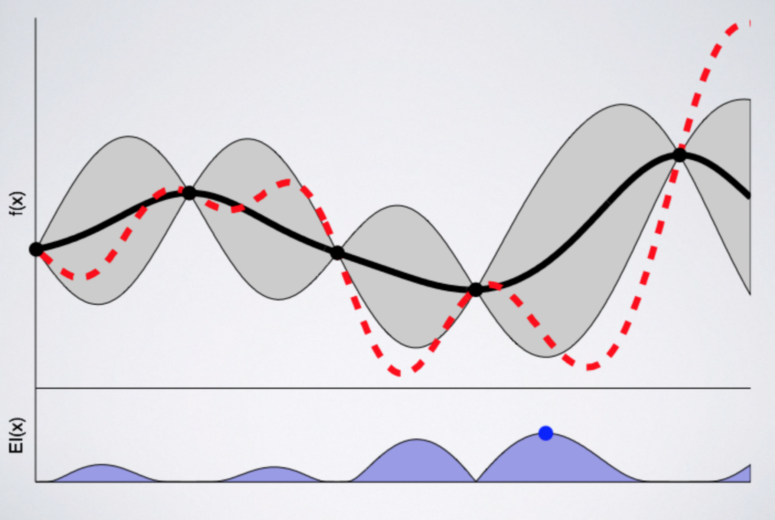
Illustrating Bayesian Optimization



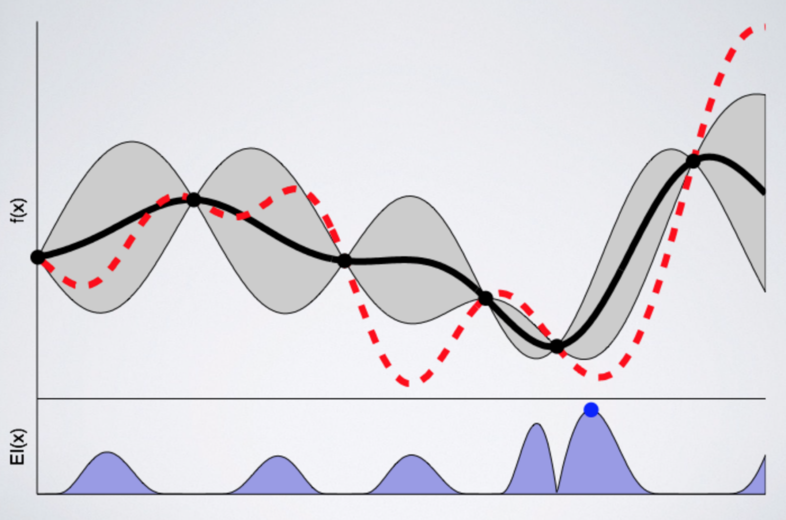
Illustrating Bayesian Optimization



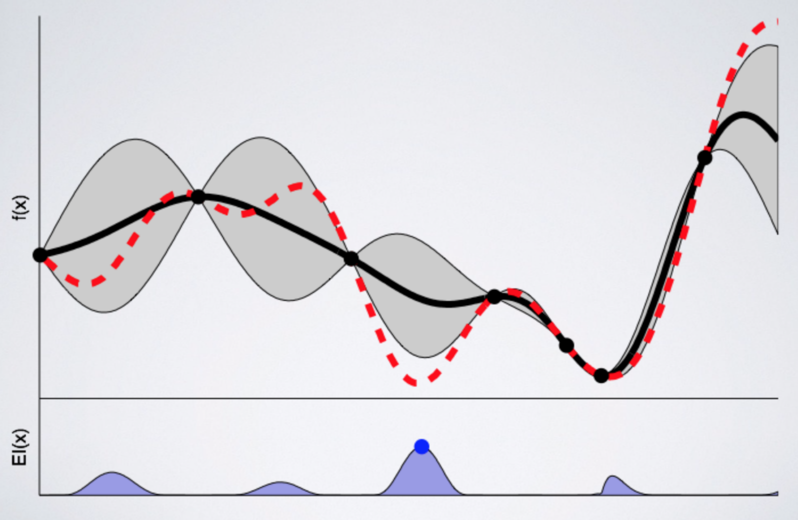
Illustrating Bayesian Optimization



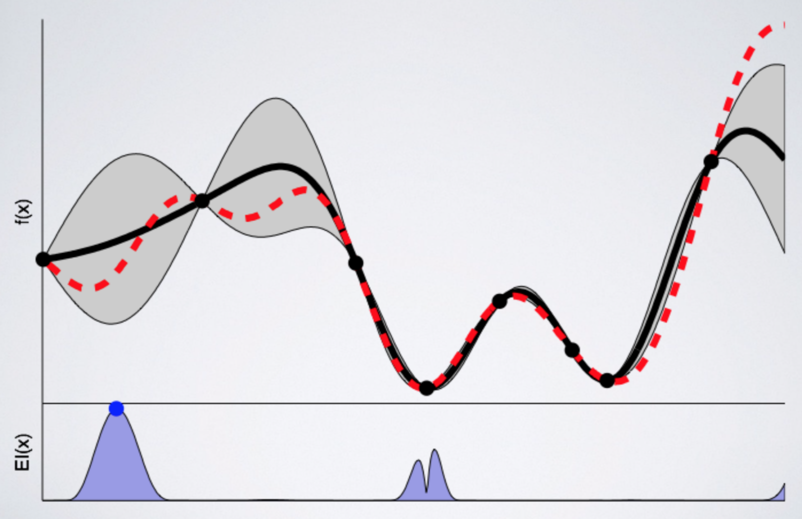
Illustrating Bayesian Optimization



Illustrating Bayesian Optimization



Illustrating Bayesian Optimization

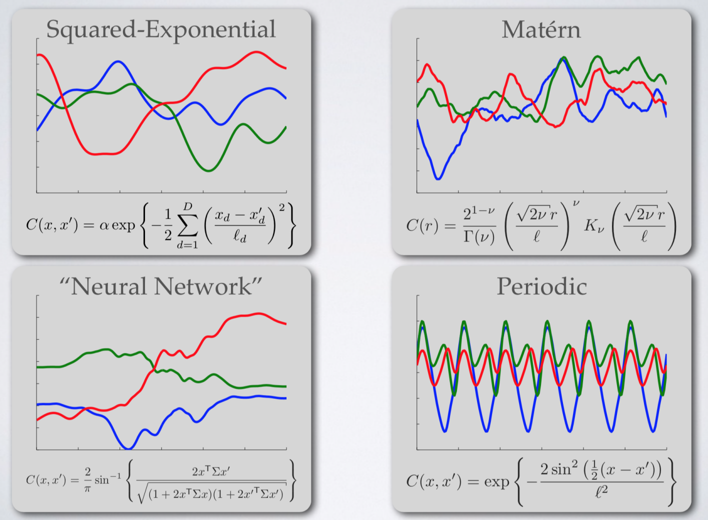


Images taken from Ryan P. Adams's Slides.

Downsides

- Experiments are run sequentially. We want to take advantage of cluster computing.
- Limited scalability in dimensions and evaluations.
- Bayesian optimization has its own hyperparameters.
- Covariance function selection.

Kernels



$$C(x,x') = \alpha \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - x'_d}{\ell_d}\right)^2\right\}$$

$$C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,r}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}\,r}{\ell}\right)$$

$$C(x,x') = \frac{2}{\pi}\sin^{-1}\left\{\frac{2x^{\mathsf{T}}\Sigma x'}{\sqrt{(1+2x^{\mathsf{T}}\Sigma x)(1+2x'^{\mathsf{T}}\Sigma x')}}\right\}$$

$$C(x,x') = \exp\left\{-\frac{2\sin^2\left(\frac{1}{2}(x-x')\right)}{\ell^2}\right\}$$

Image taken from Ryan P. Adams's Slides.