Tutorial: Bayesian Methods for Global and Simulation Optimization

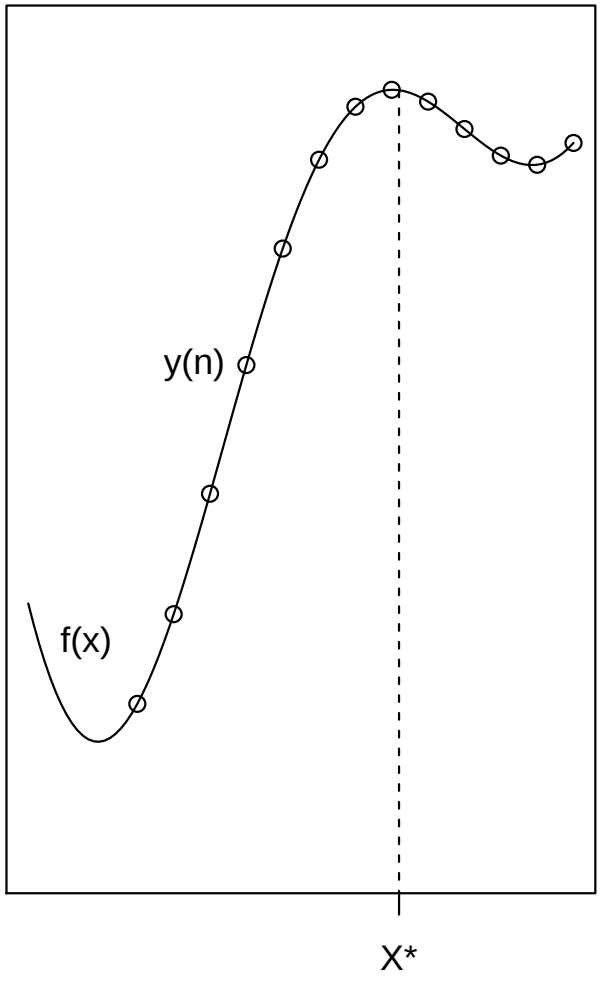
Peter I. Frazier

Operations Research & Information Engineering, Cornell University

Monday November 14, 2011
INFORMS Annual Meeting
Charlotte, NC

Outline

Noise-Free Global Optimization



- Objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$, continuous but generally not concave.
- Feasible set $A \subseteq \mathbb{R}^d$.
- Our goal is to solve

$$\max_{x \in A} f(x)$$

- Typically, f is time-consuming to evaluate, derivative information is unavailable, and the dimension is not too large ($d < 20$).

Noise-Free Global Optimization Has Lots of Applications

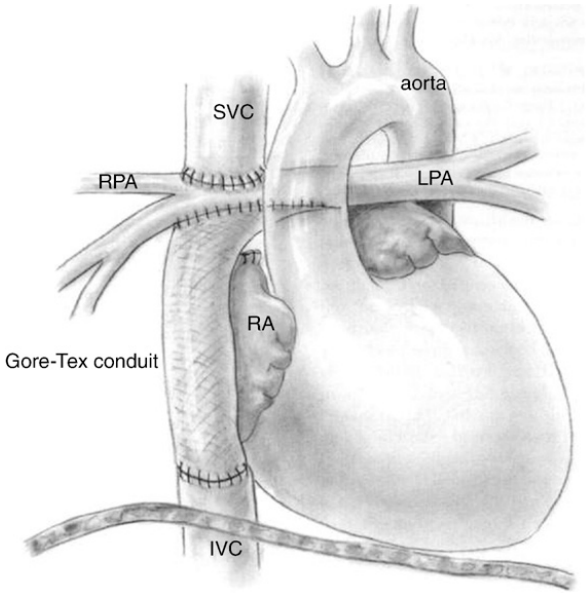
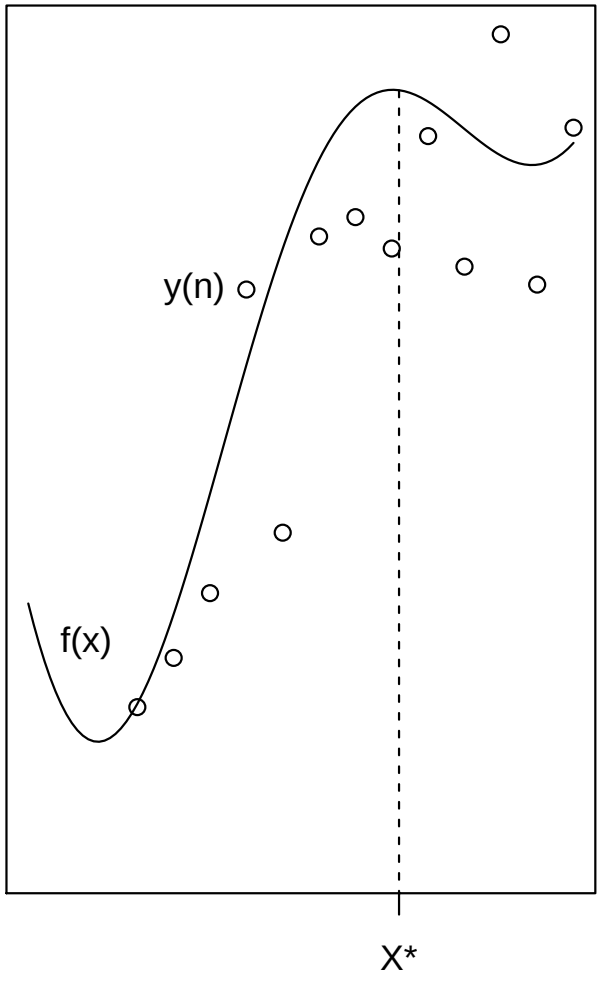


Fig. 1. Extracardiac total cavopulmonary connection. The IVC is disconnected from the right atrium (RA) and connected to the PAs via a Gore-Tex conduit. Figure taken from Reddy et al. [13].

- Design of grafts to be used in heart surgery. [Yang et al., 2010]
- Design of aeorodynamic structures, e.g., cars, airplanes. [Forrester et al., 2008]
- Calibrating the parameters of a climate model to historical data.
- Tuning the parameters of software for assembling short DNA reads into genomes. (current project).

Noisy Global Optimization

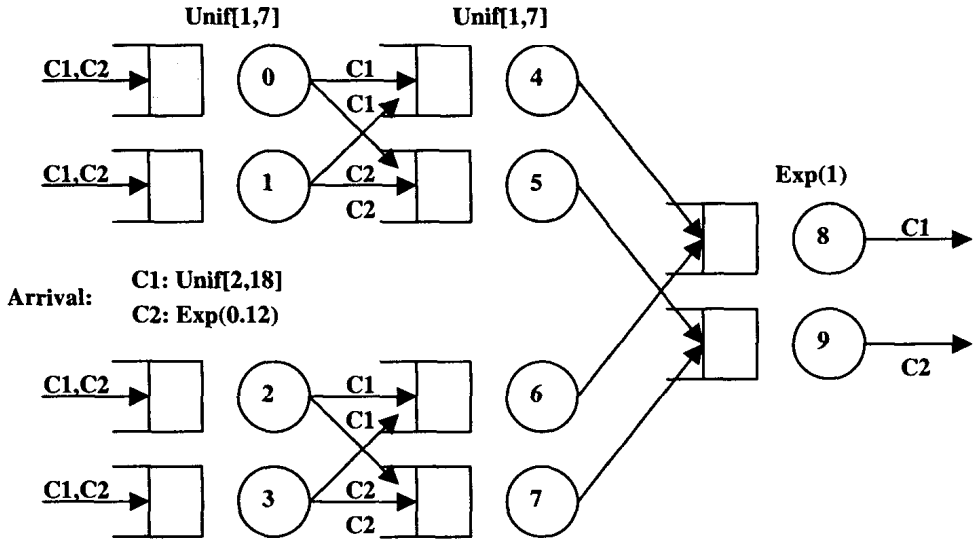


- We cannot evaluate $f(x)$ directly.
- Instead, we have a stochastic simulator that can evaluate $f(x)$ with noise.
- It gives us $g(x,\omega) = f(x) + \varepsilon(x,\omega)$, where $E[g(x,\omega)] = f(x)$.
- Our goal is still to find a global maximum,

$$\max_{x \in A} f(x)$$

- The term **simulation optimization** is also used.

Noisy Global Optimization Has Lots of Applications



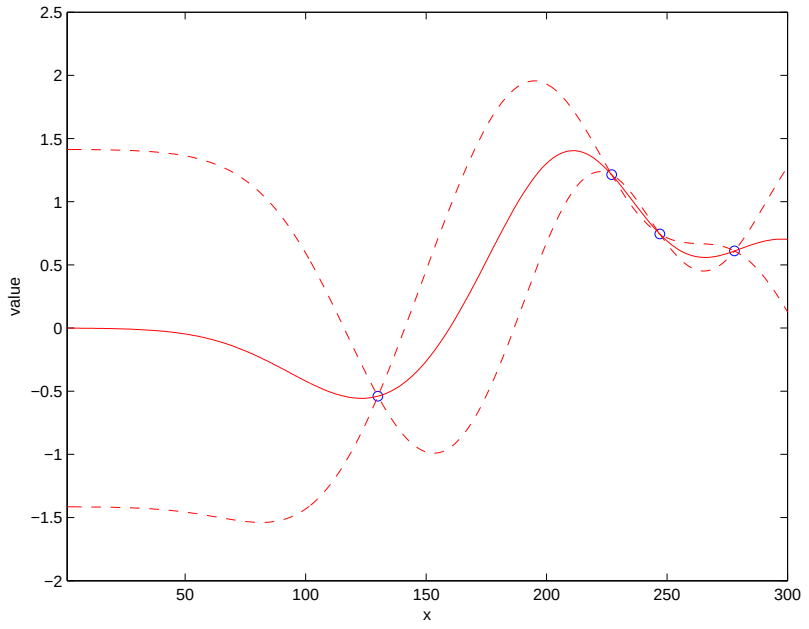
- Choose staffing levels in a hospital, using a discrete-event simulator.
- Choose an admissions control policy in a complex queuing system, e.g., a call center.
- Calibrate a logistics model to historical data (case study).
- Drug development (case study).

What is Bayesian Global Optimization?

- Bayesian Global Optimization (BGO) is a class of algorithms for solving Noise-Free and Noisy Global Optimization problems.
- These algorithms use methods from Bayesian statistics to decide where to sample.

BGO uses Bayesian Statistics to Decide Where to Sample

- Given the function evaluations obtained so for, a BGO algorithm uses Bayesian methods to get:
  - estimates of $f(x)$ over the feasible set.
  - uncertainties in these estimates.
  - together, these are described by the **posterior distribution**.

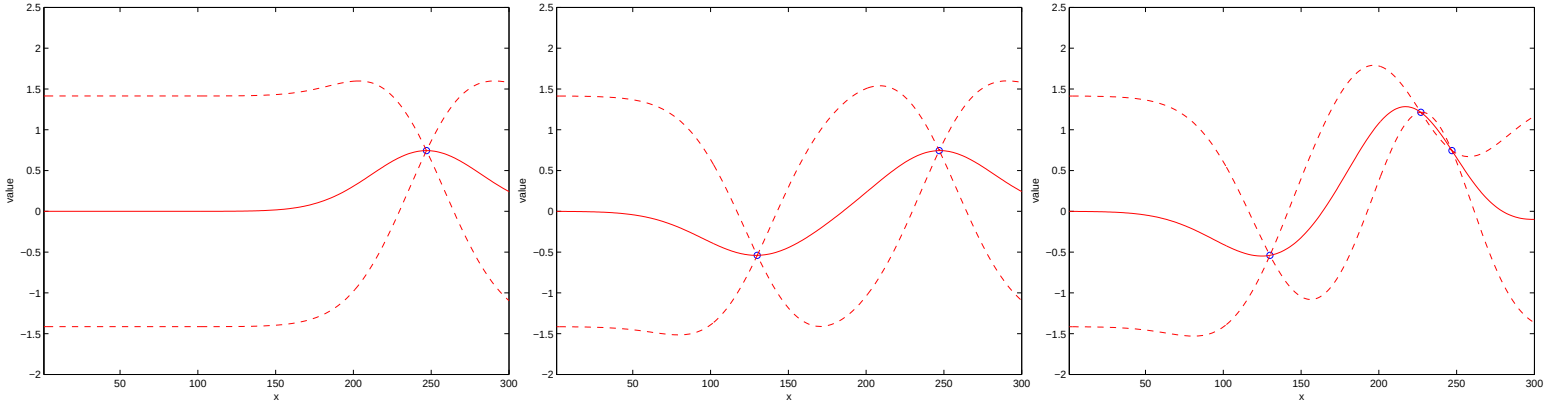


- BGO uses the posterior distribution to decide where to evaluate next.

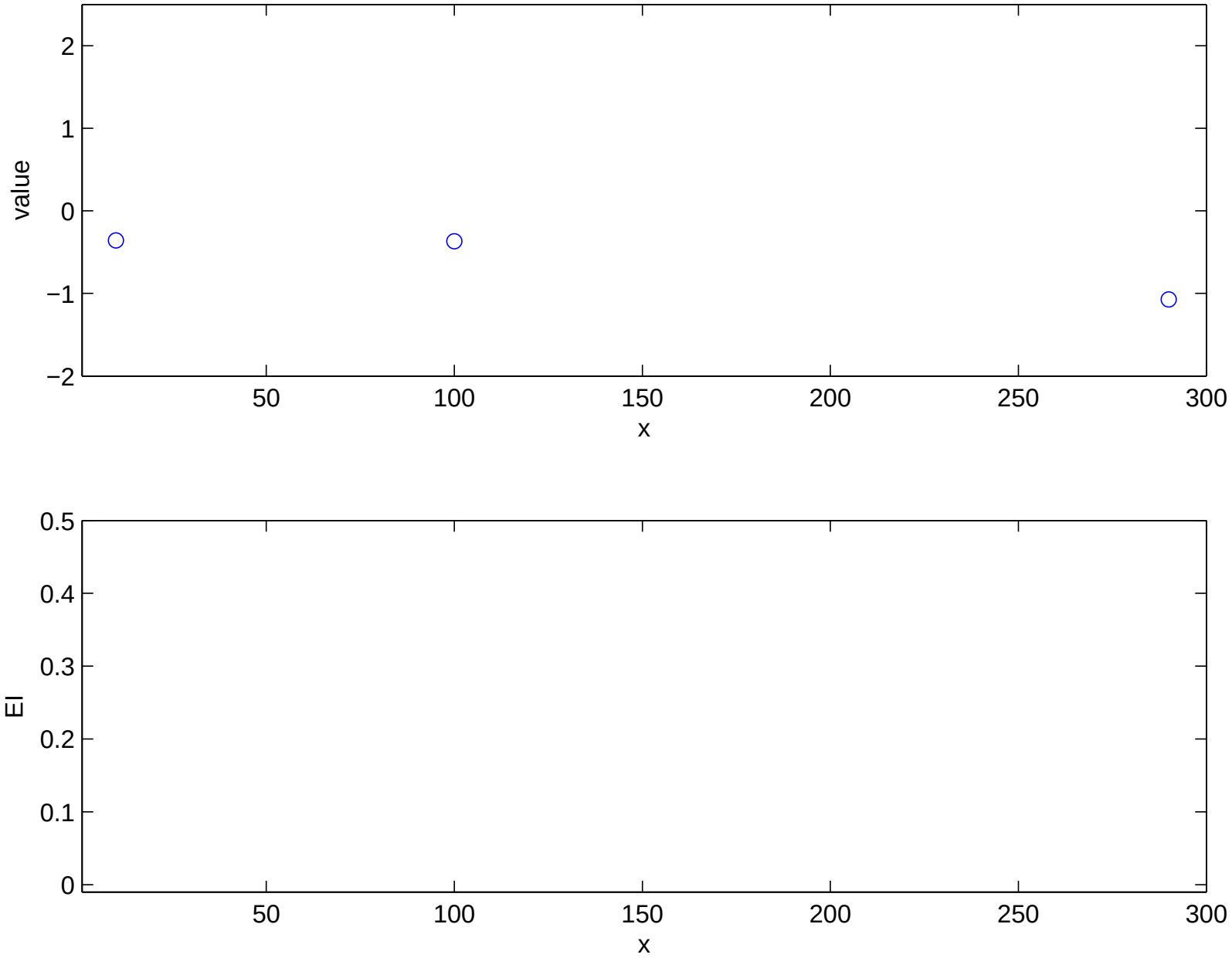
Typical BGO Algorithm

1. Choose several initial points x and evaluate $f(x)$ or $g(x, \omega)$.
2. While the stopping criterion is not met:
   - 2a. Calculate the Bayesian posterior distribution on f from the points observed.
   - 2b. Use the posterior to decide where to evaluate next.
3. Based on the most recent posterior distribution, report the point with the best estimated value.

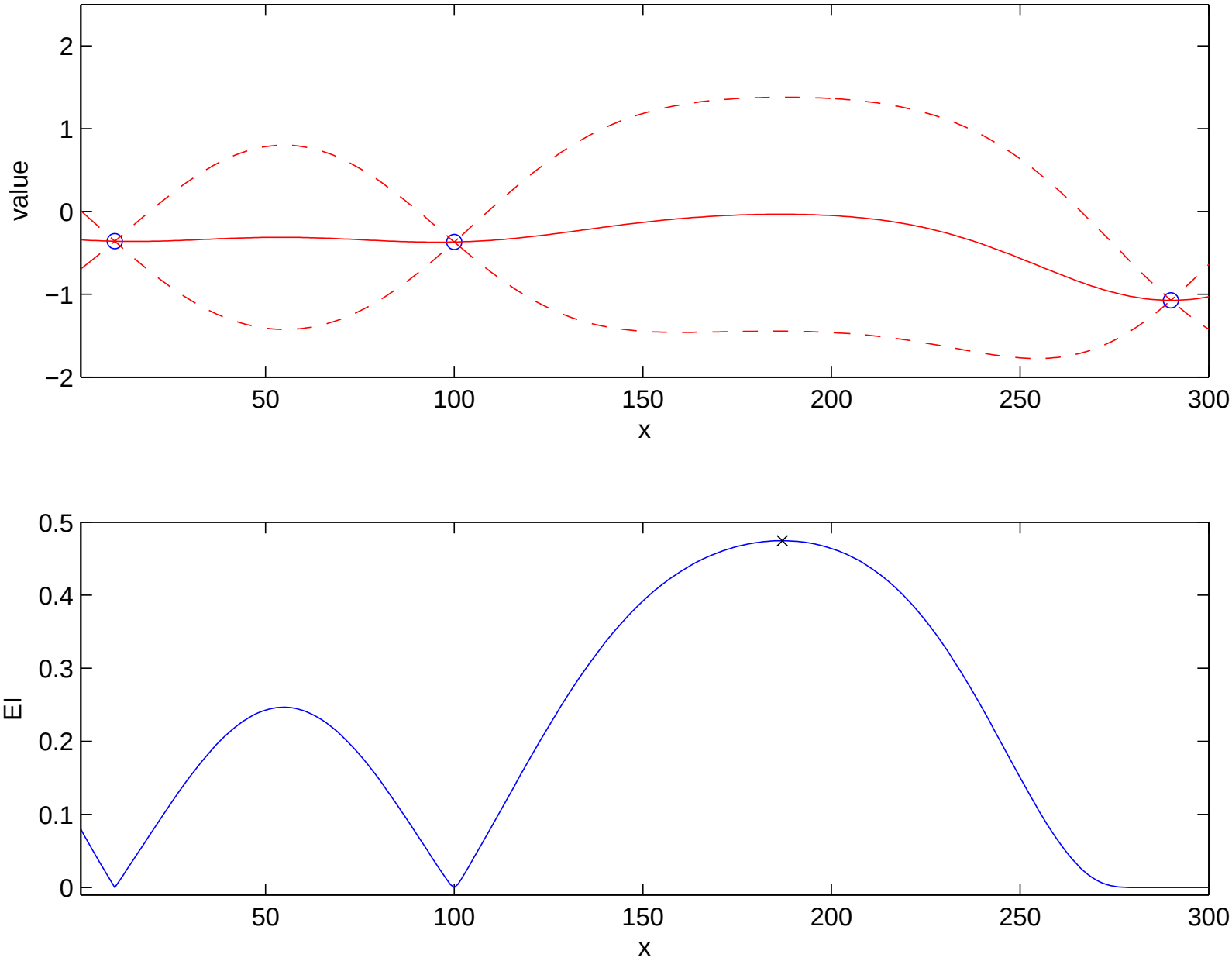
The stopping criteria is often "stop after N samples", but can be more sophisticated.



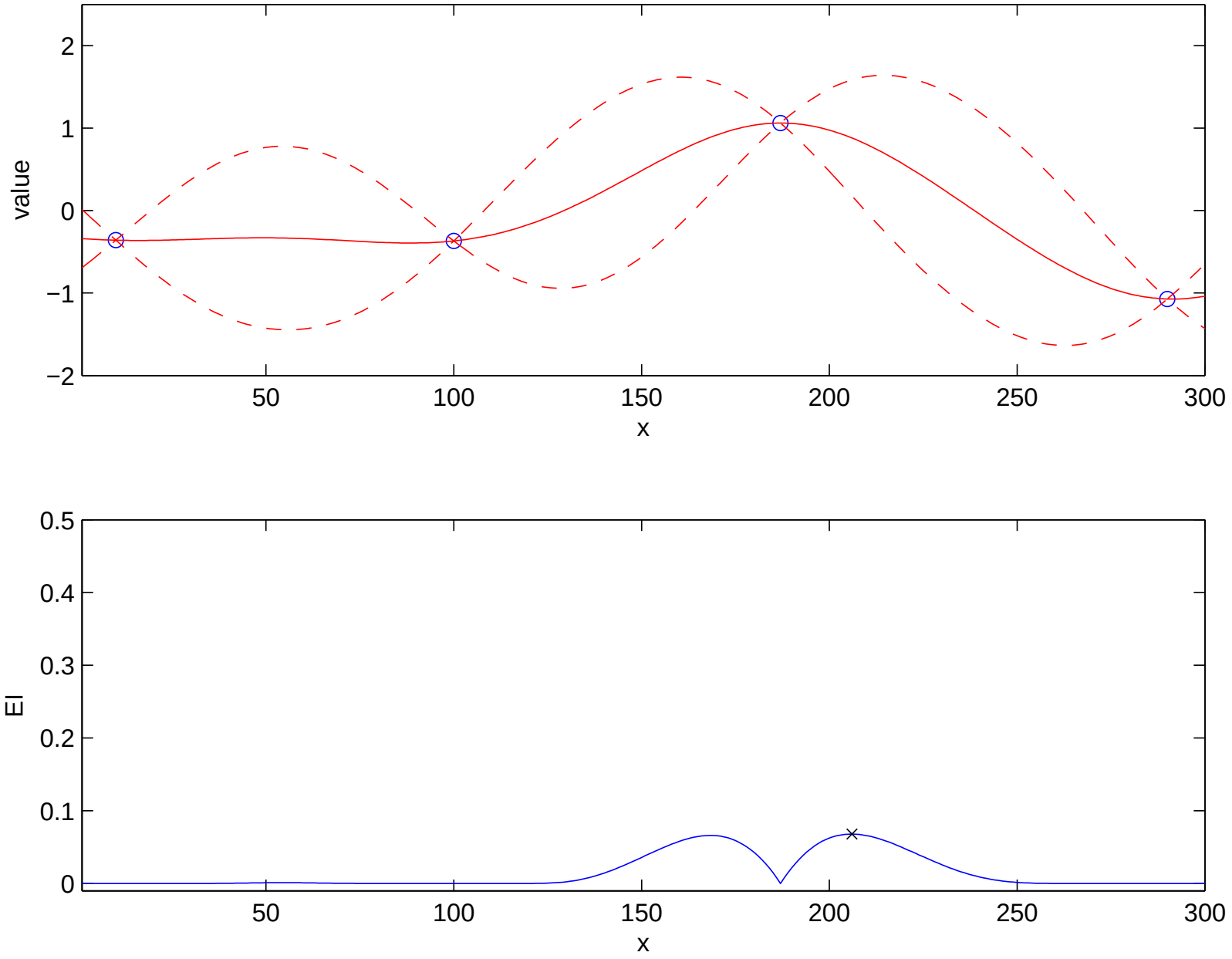
Animation of a BGO Algorithm



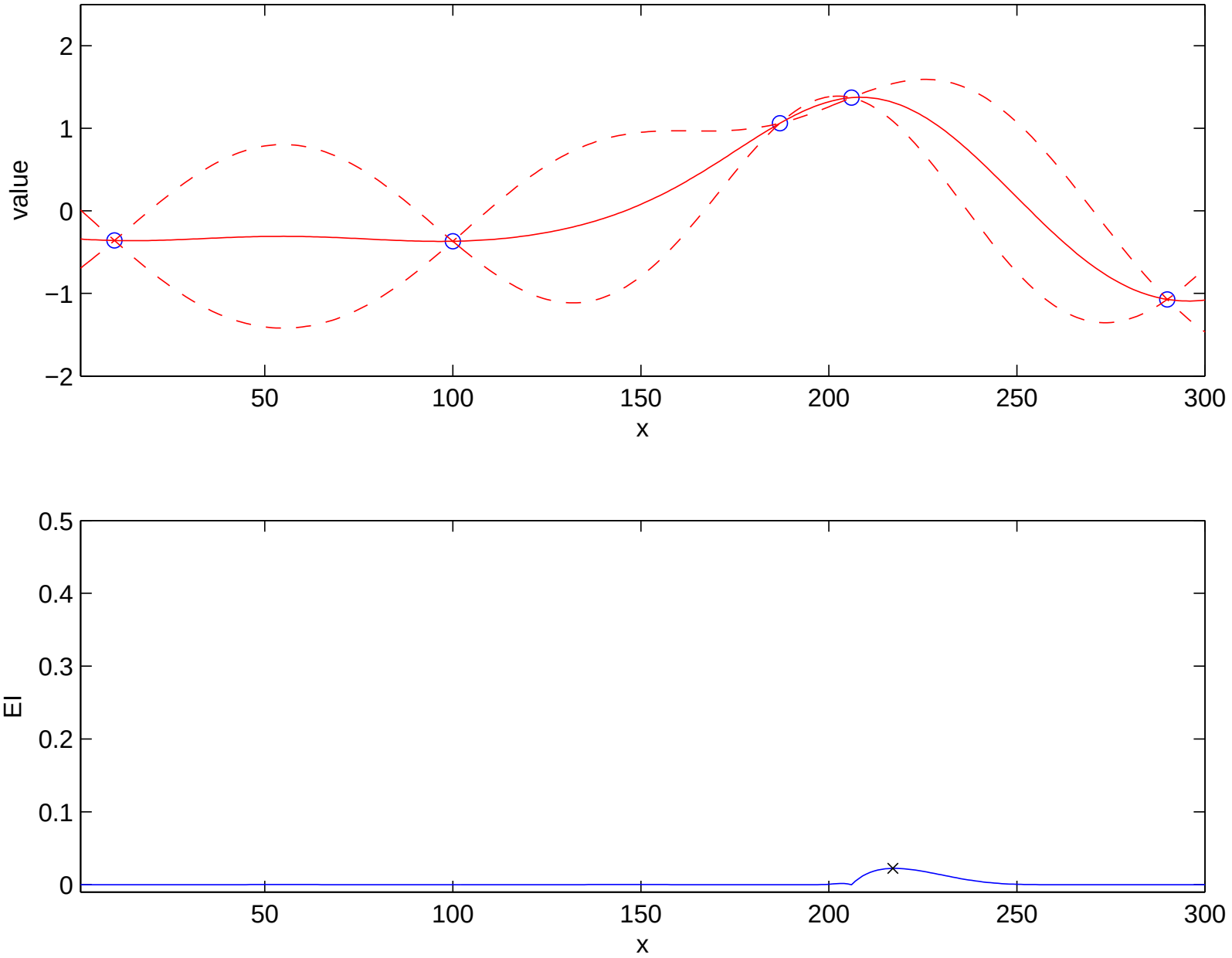
Animation of a BGO Algorithm



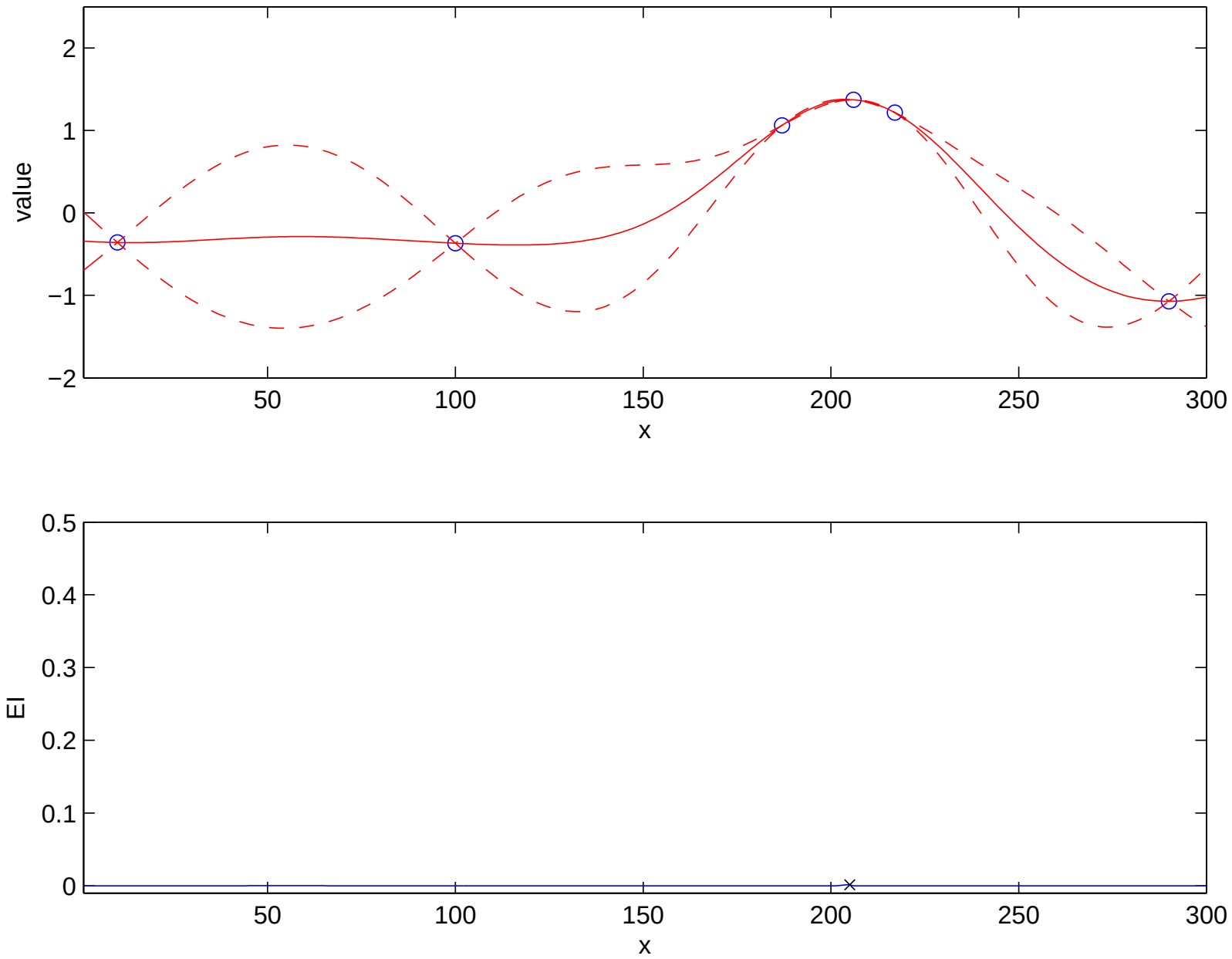
Animation of a BGO Algorithm



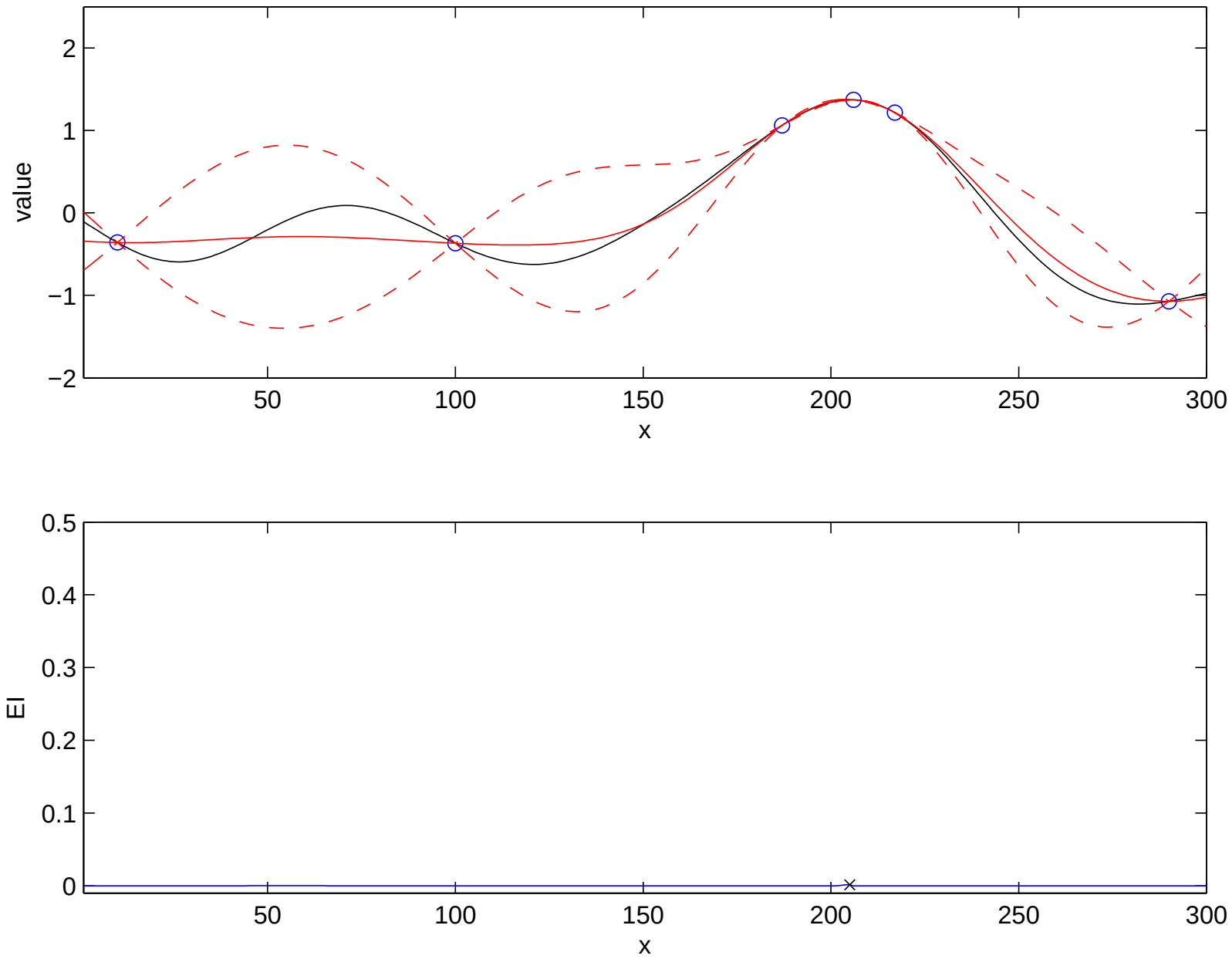
Animation of a BGO Algorithm



Animation of a BGO Algorithm



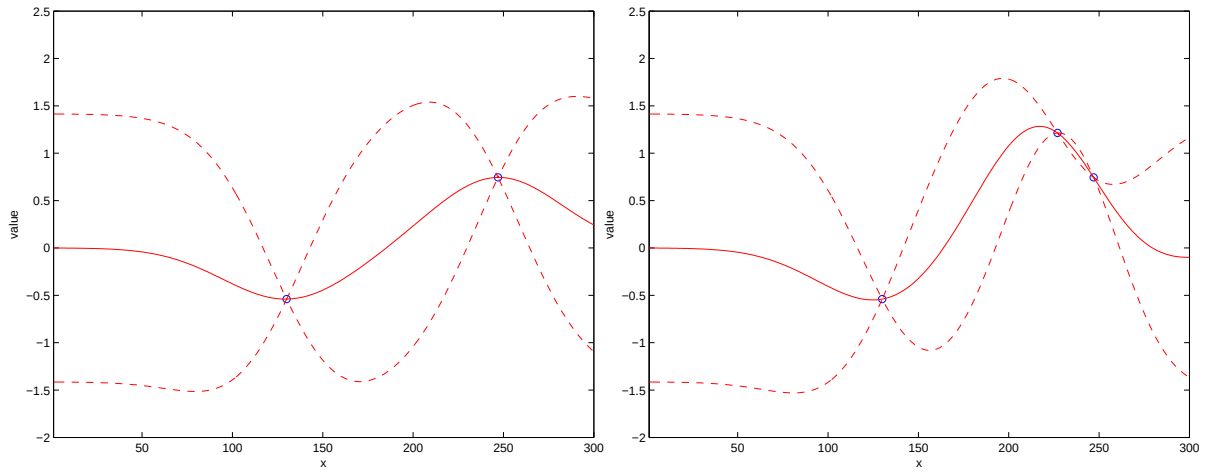
Animation of a BGO Algorithm



Outline

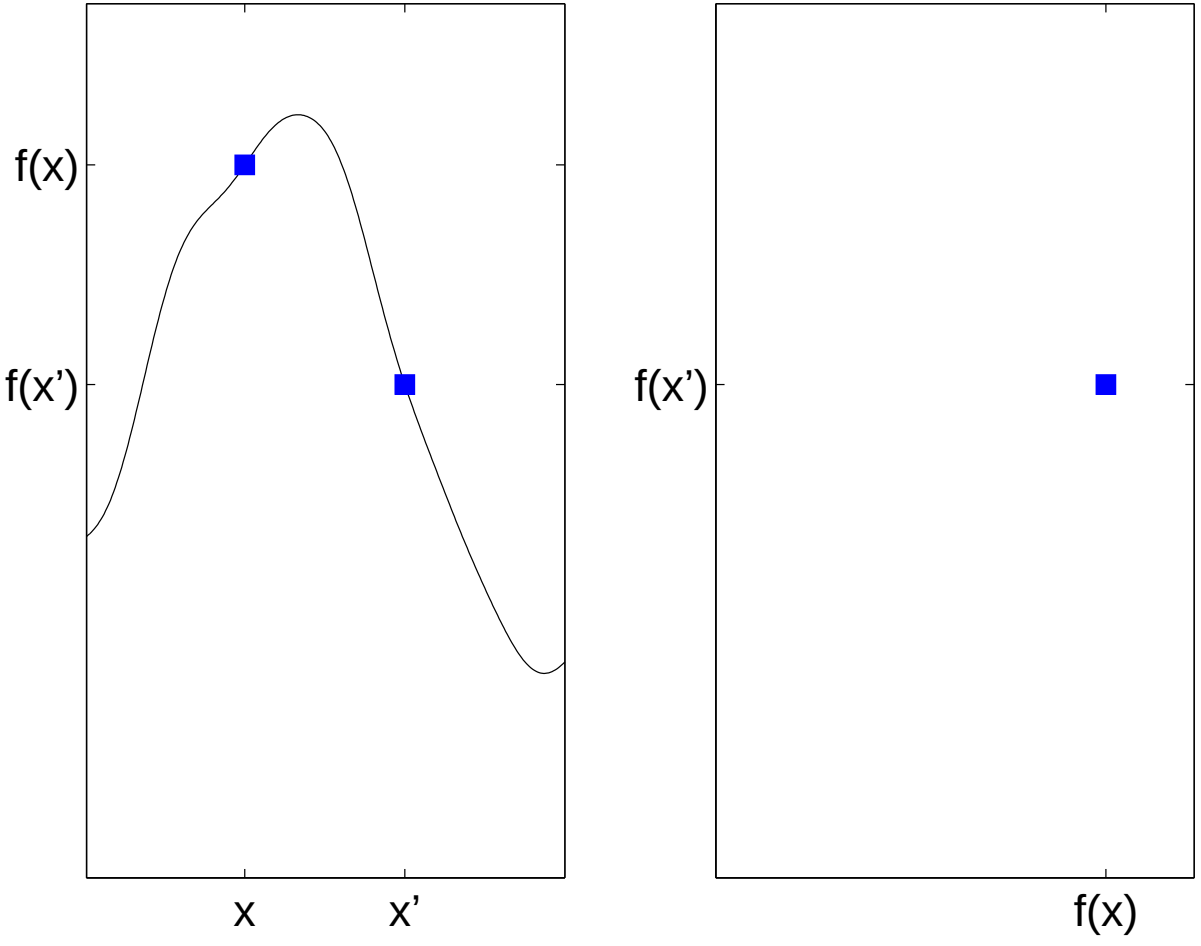
Illustration of Gaussian Process (GP) Regression

- Left: 2 noise-free function evaluations (blue), estimate of f (solid red), confidence bounds on this estimate (dashed red).
- Right: One more function evaluation is added.



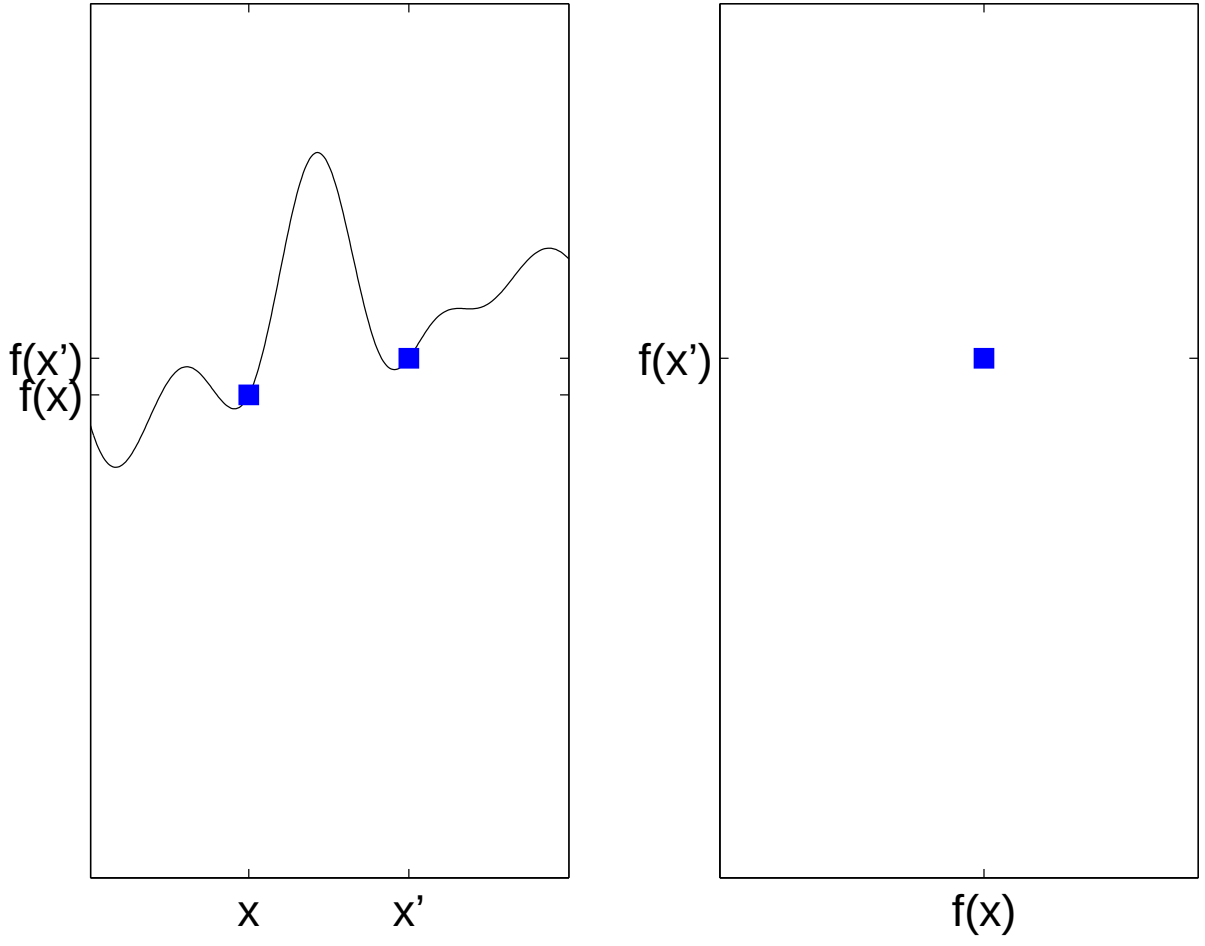
Gaussian Process Regression: Two Points

- Fix two points x and x'.
- Consider the values of f at these points, $f(x)$ and $f(x')$.



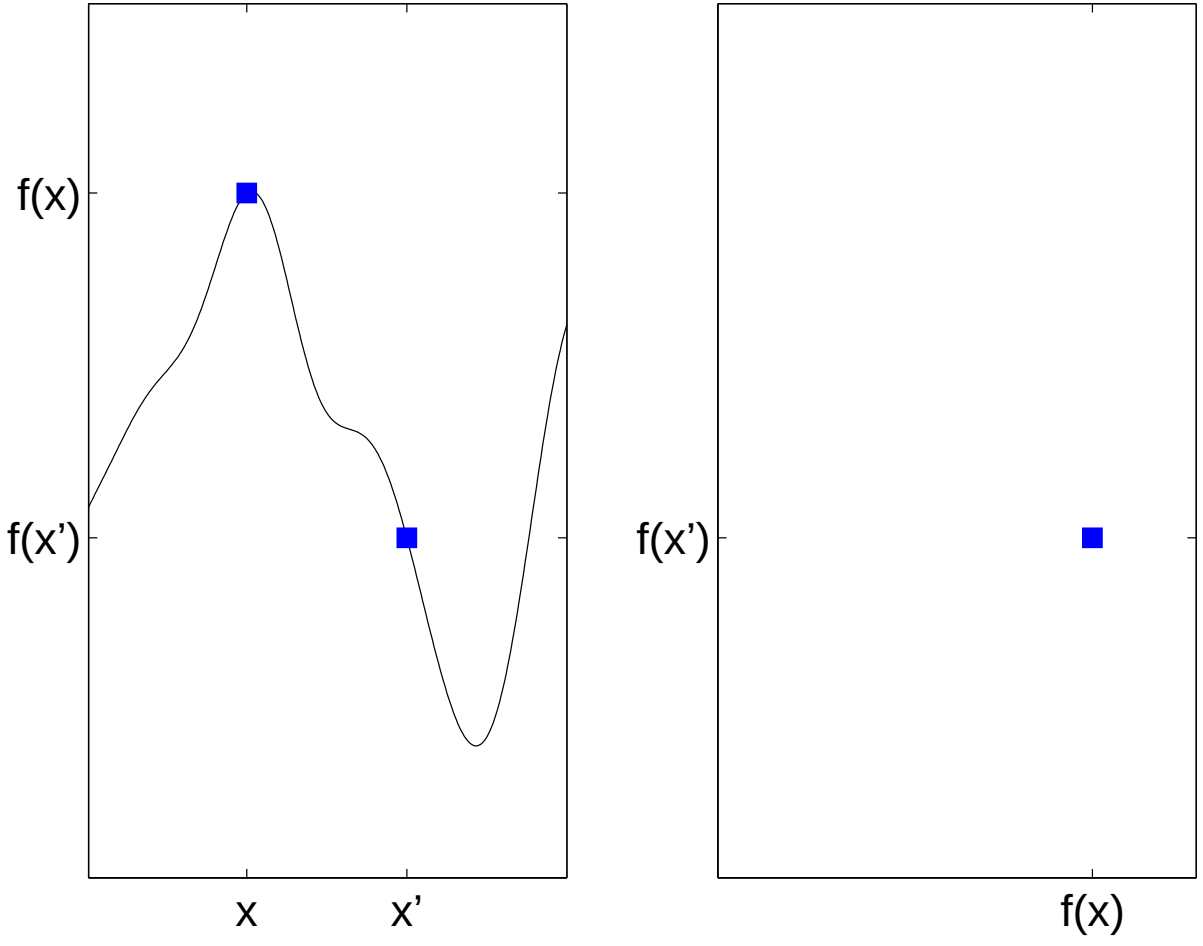
Gaussian Process Regression: Two Points

- Fix two points x and x'.
- Consider the values of f at these points, $f(x)$ and $f(x')$.



Gaussian Process Regression: Two Points

- Fix two points x and x'.
- Consider the values of f at these points, $f(x)$ and $f(x')$.



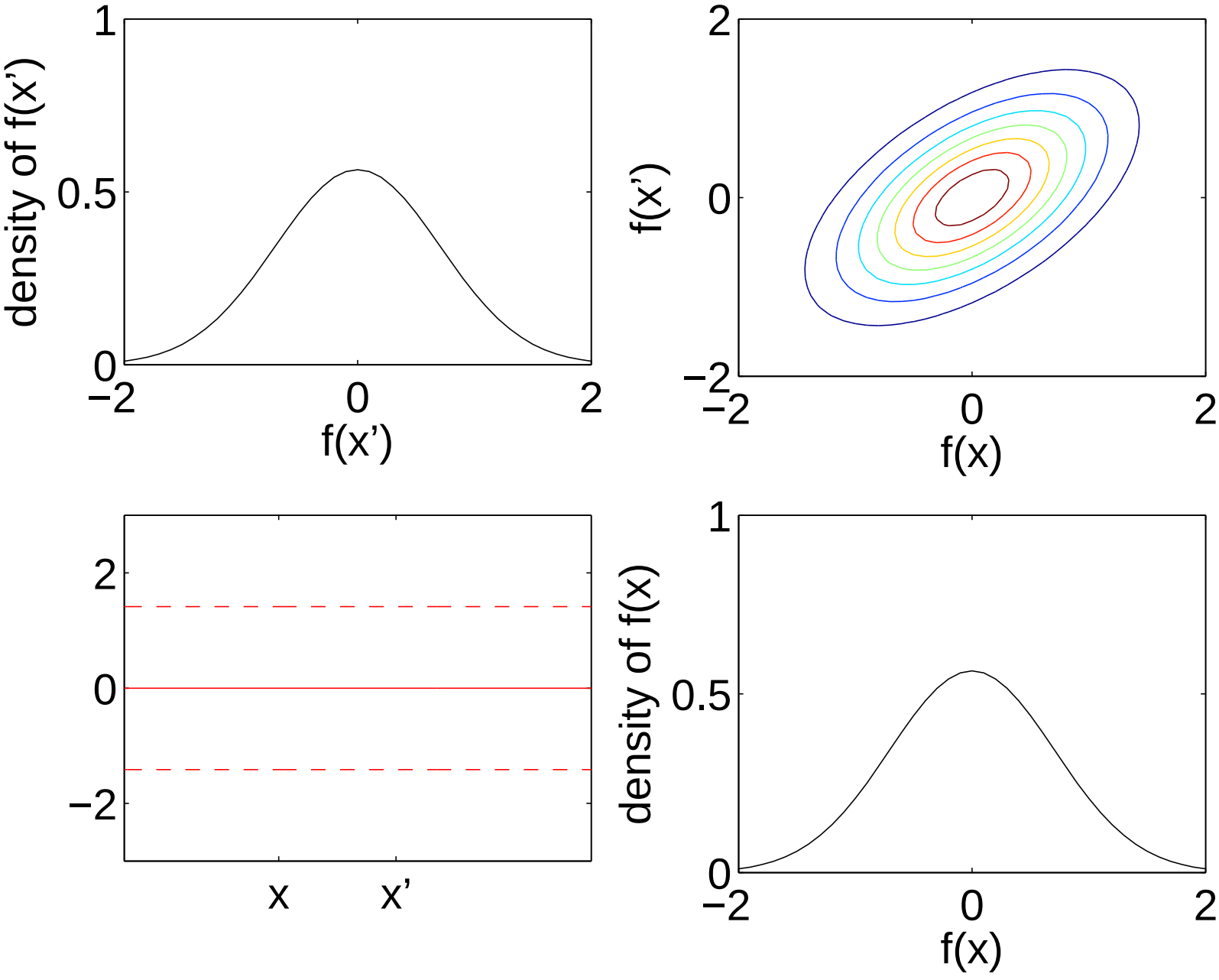
Gaussian Process Regression: Two Points

- $f(x)$ and $f(x')$ are unknown before we measure them.
- In Bayesian statistics, we model our uncertainty about f with a prior probability distribution.

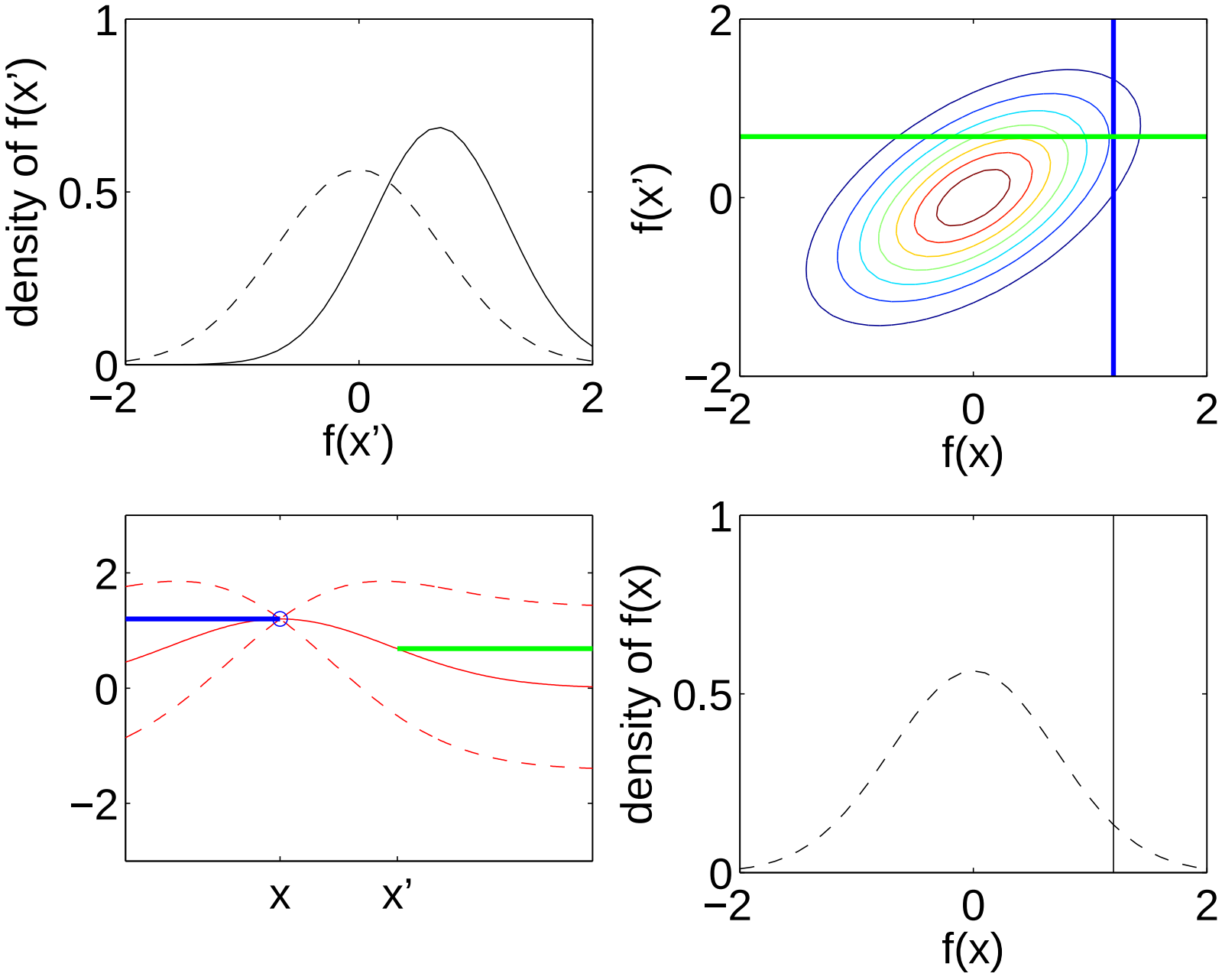
$$\begin{bmatrix} f(x) \\ f(x') \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_0(x) \\ \mu_0(x') \end{bmatrix}, \begin{bmatrix} \Sigma_0(x,x) & \Sigma_0(x,x') \\ \Sigma_0(x',x) & \Sigma_0(x',x') \end{bmatrix} \right)$$

  Here, $\mu_0(\cdot)$ and $\Sigma_0(\cdot,\cdot)$ are functions to be discussed later.
- In general, $f(x)$ and $f(x')$ are correlated.

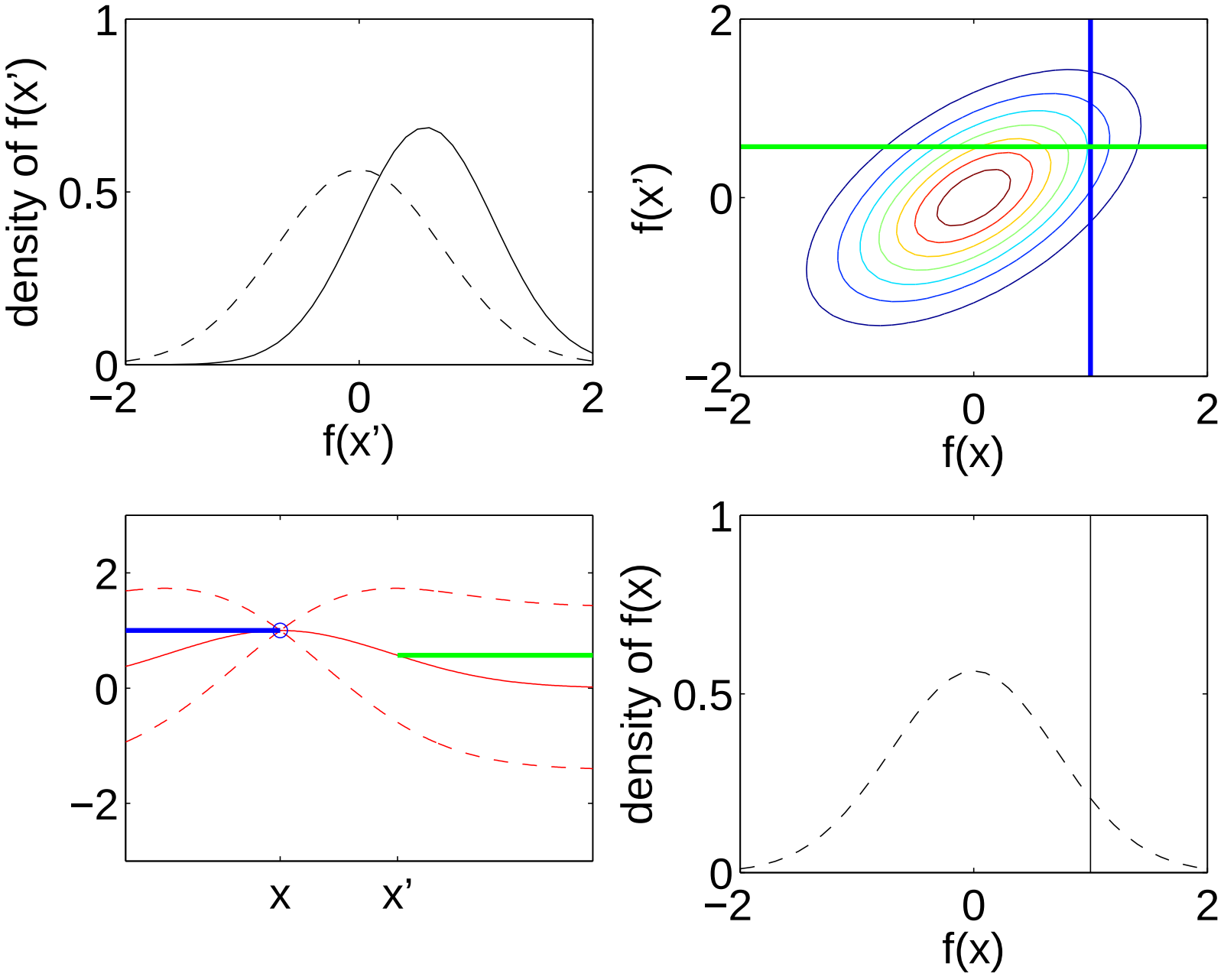
Gaussian Process Regression: Two Points



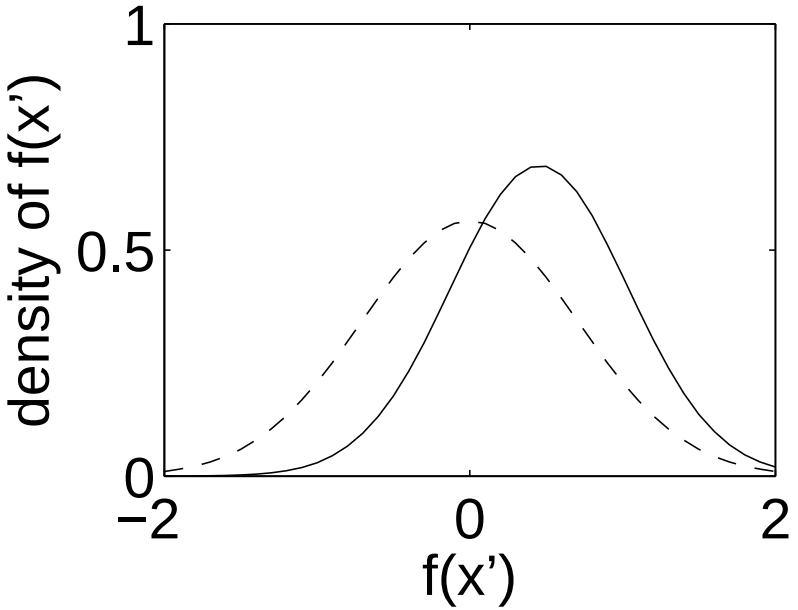
Gaussian Process Regression: Two Points

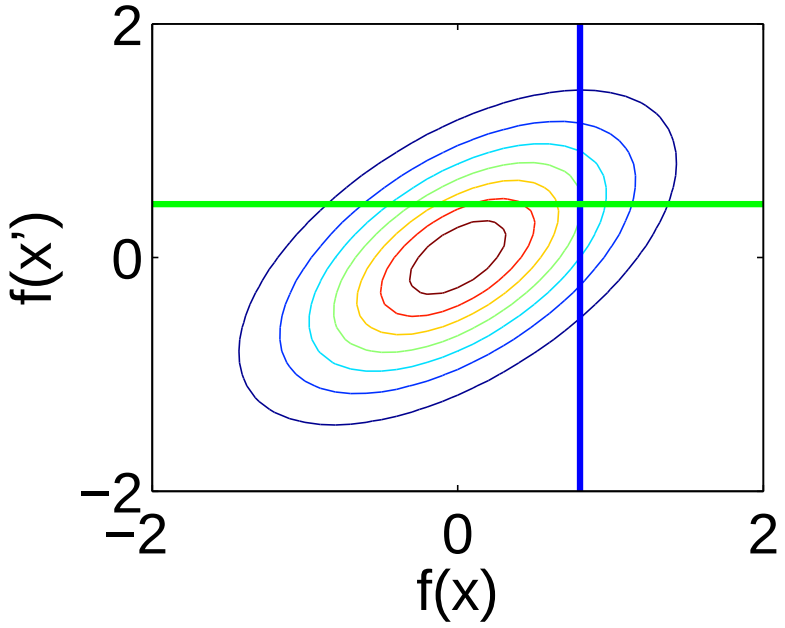


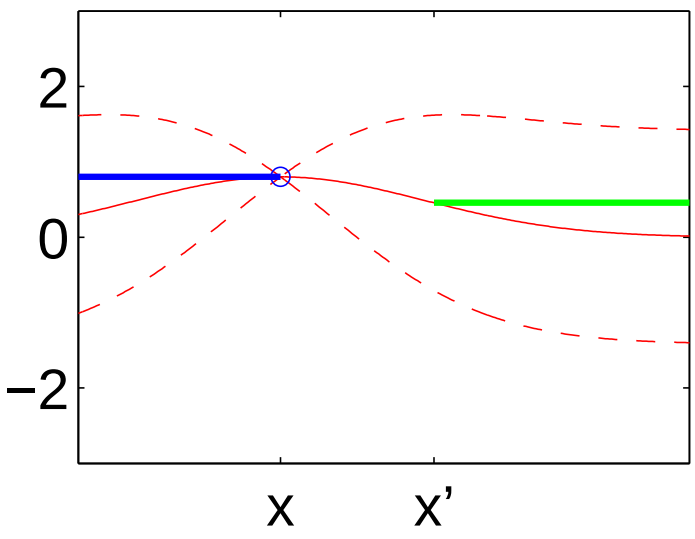
Gaussian Process Regression: Two Points

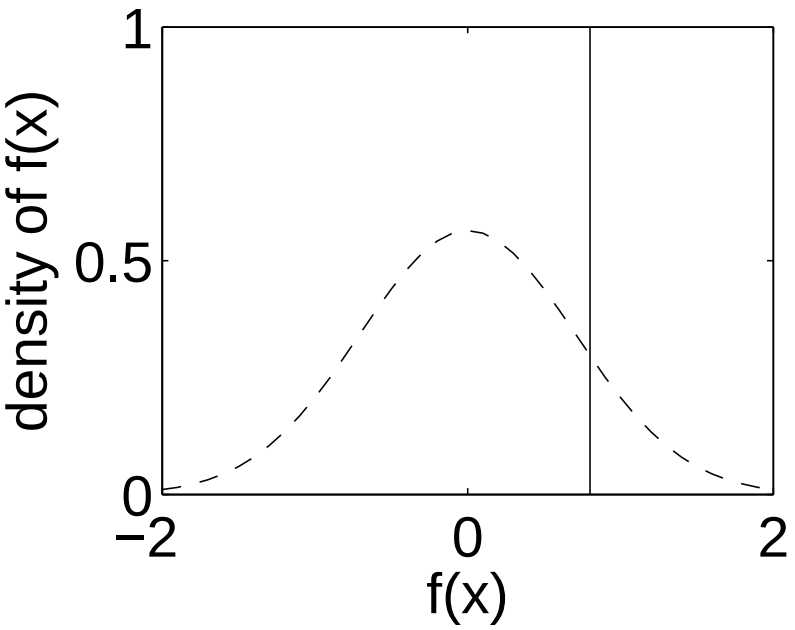


Gaussian Process Regression: Two Points

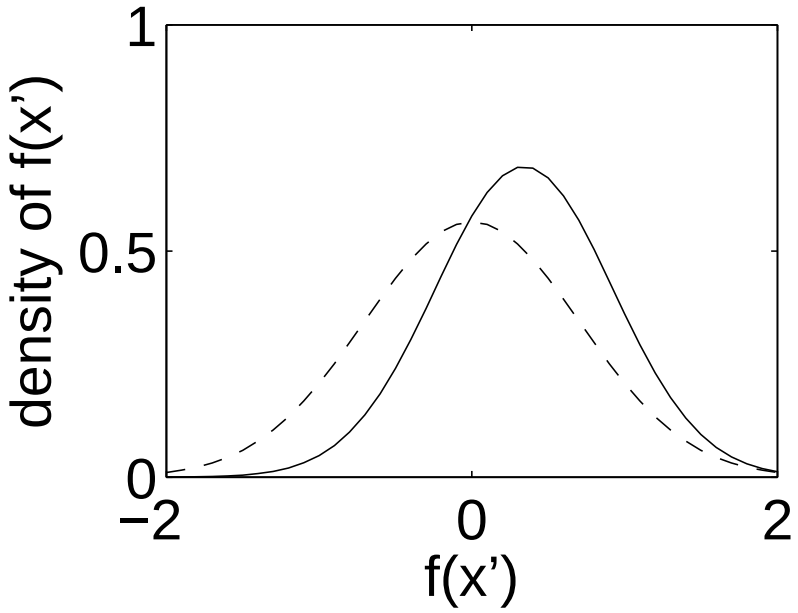


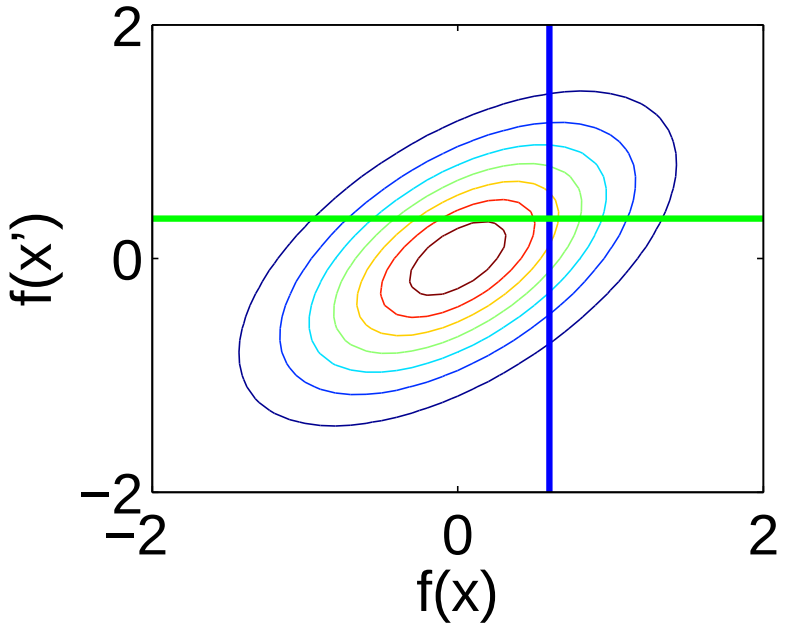


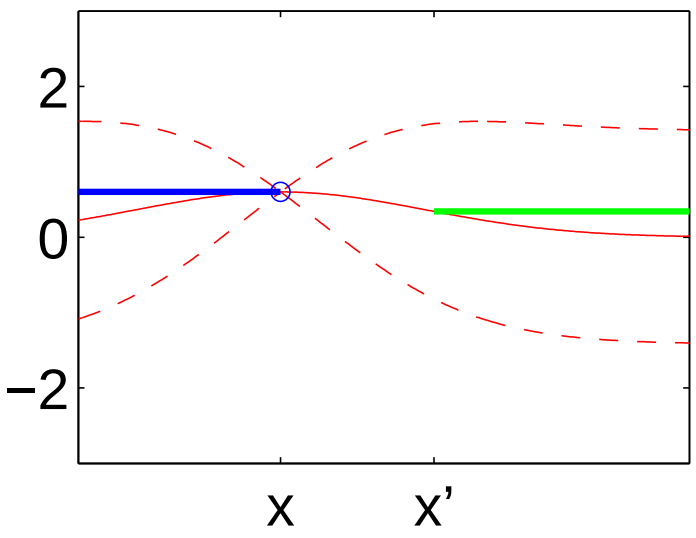


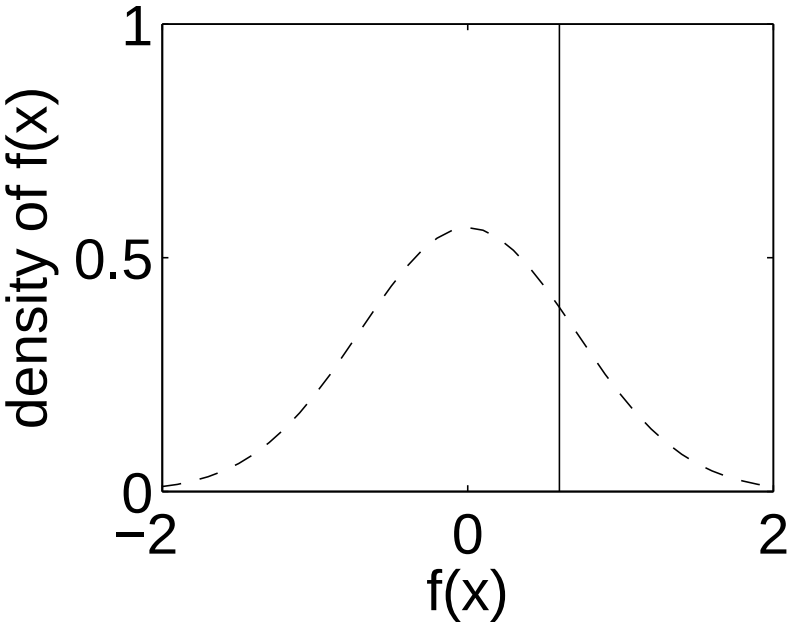


Gaussian Process Regression: Two Points

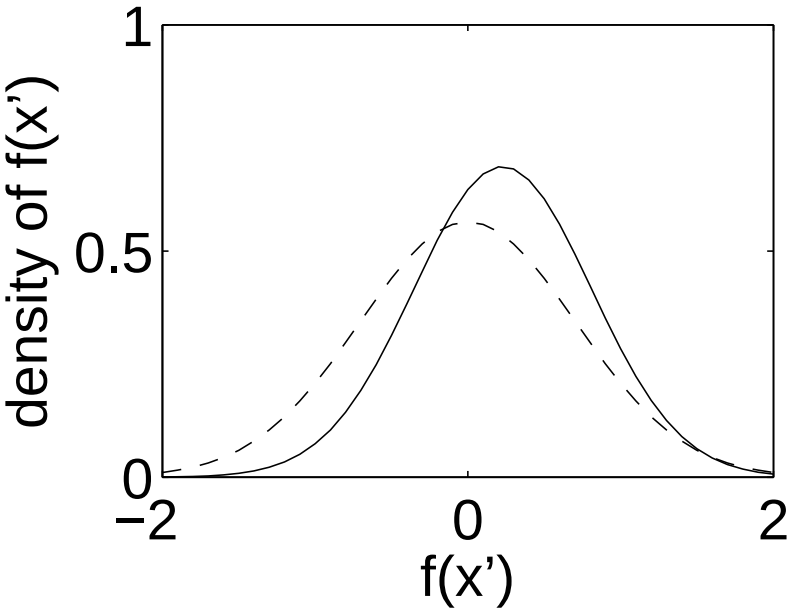


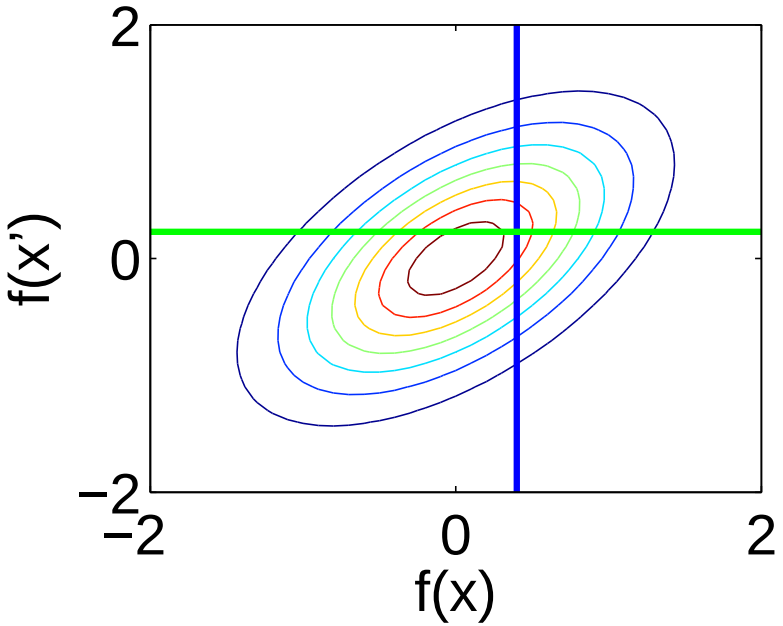


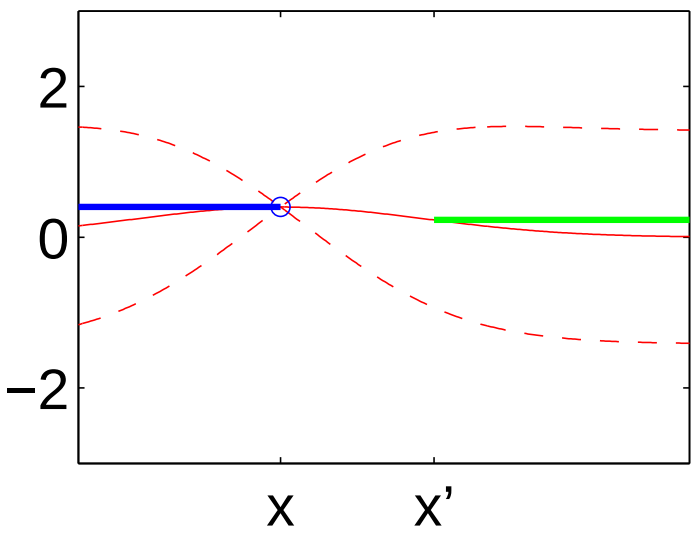


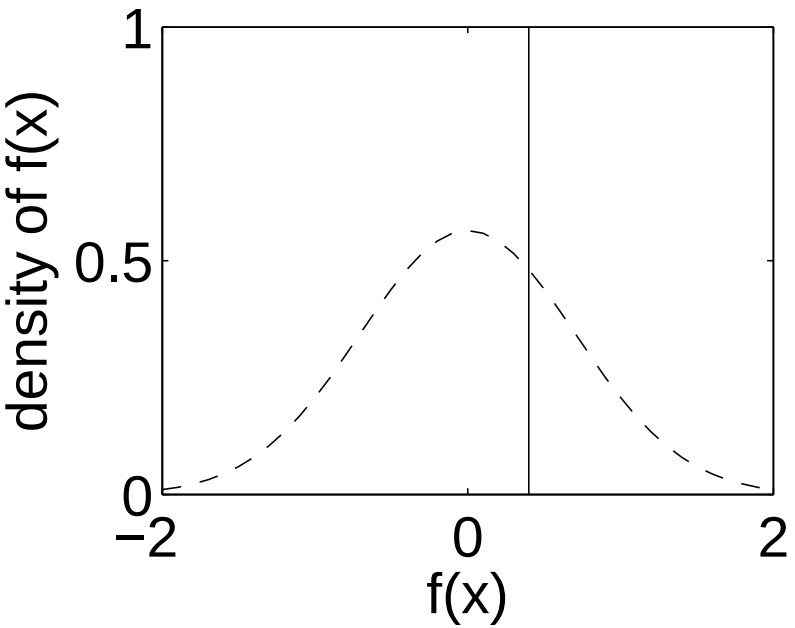


Gaussian Process Regression: Two Points

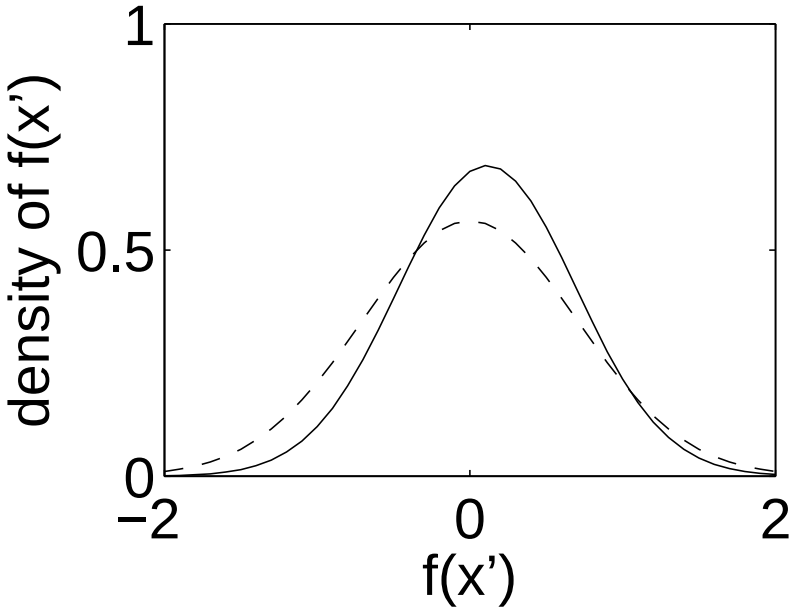


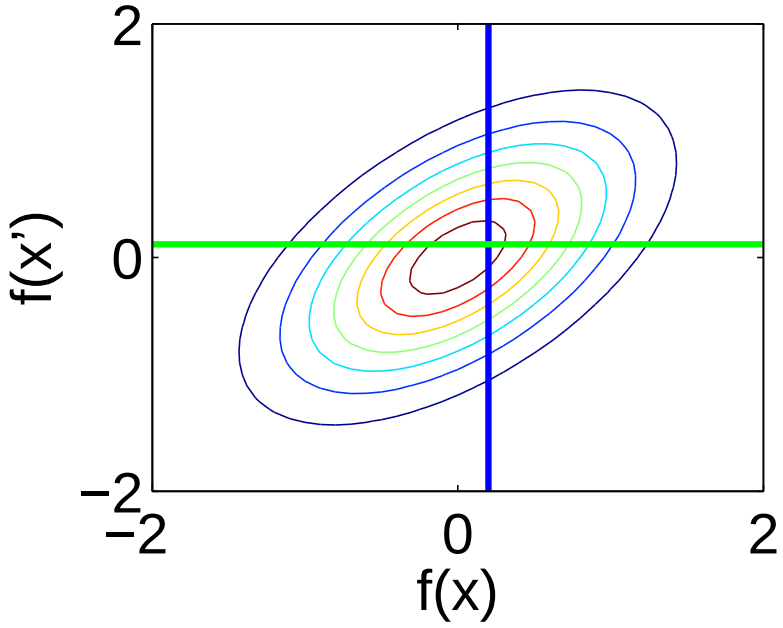


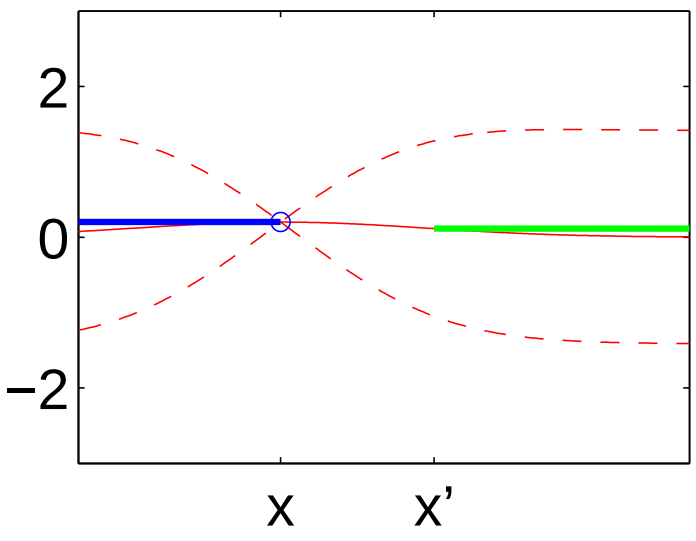


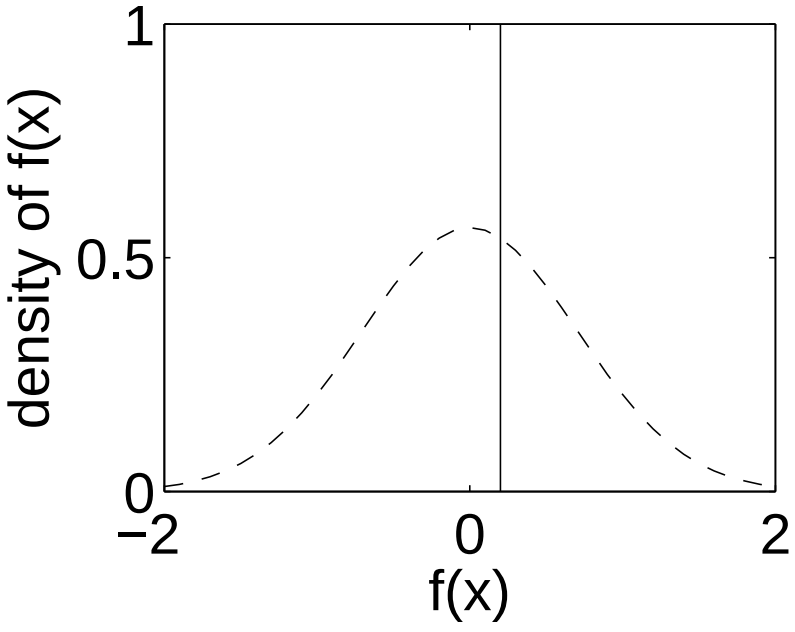


Gaussian Process Regression: Two Points

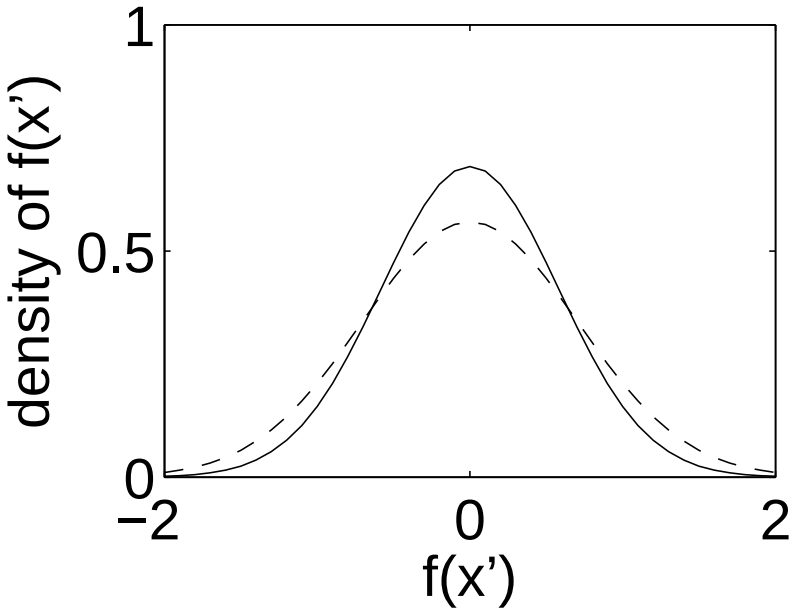


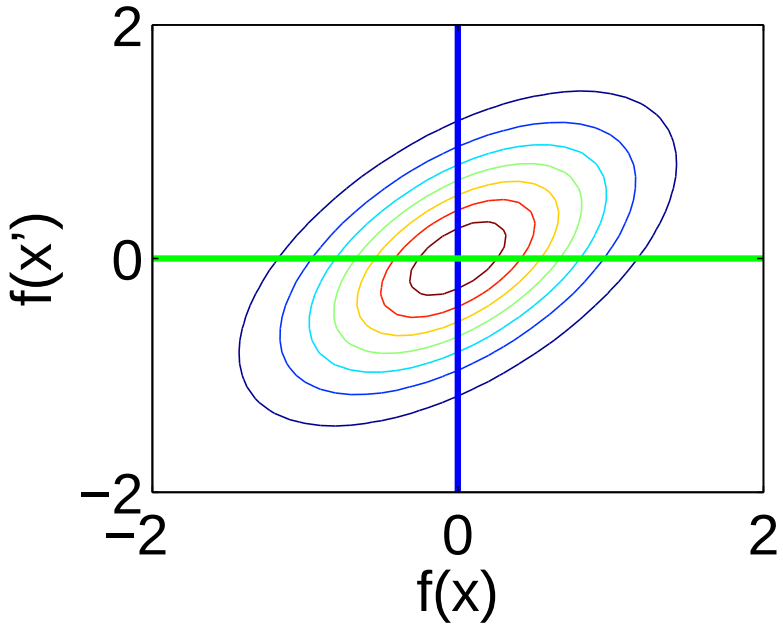


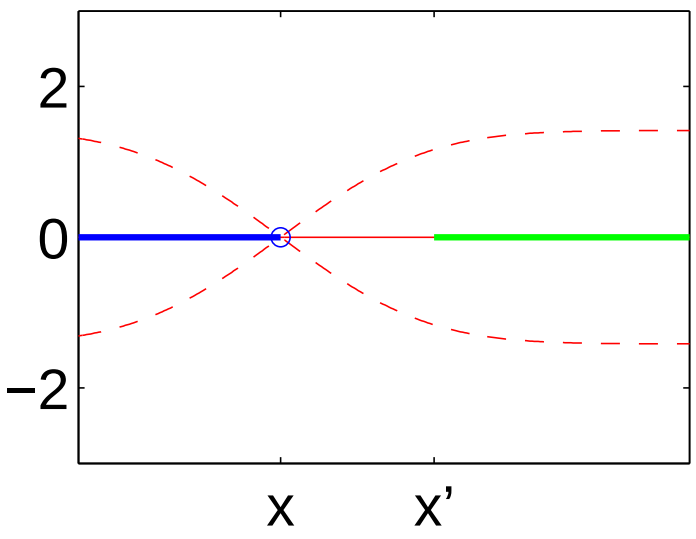


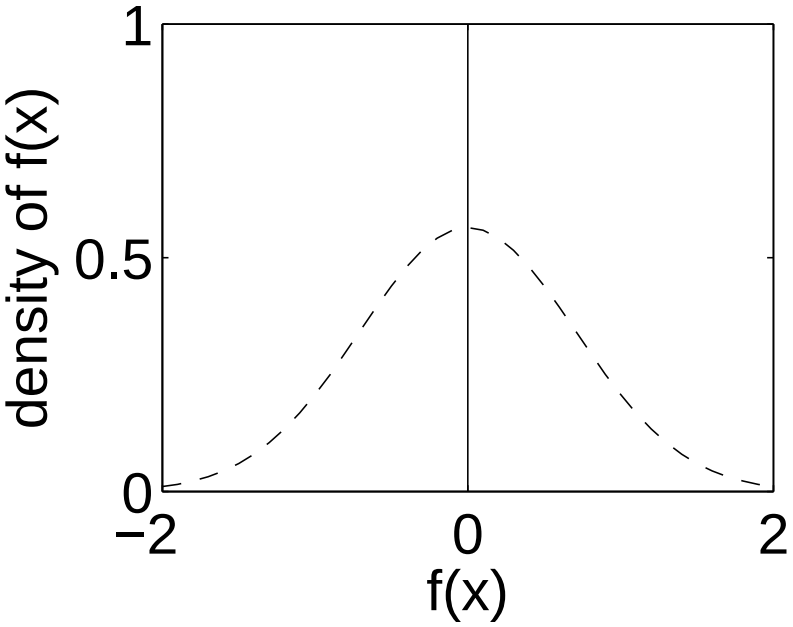


Gaussian Process Regression: Two Points

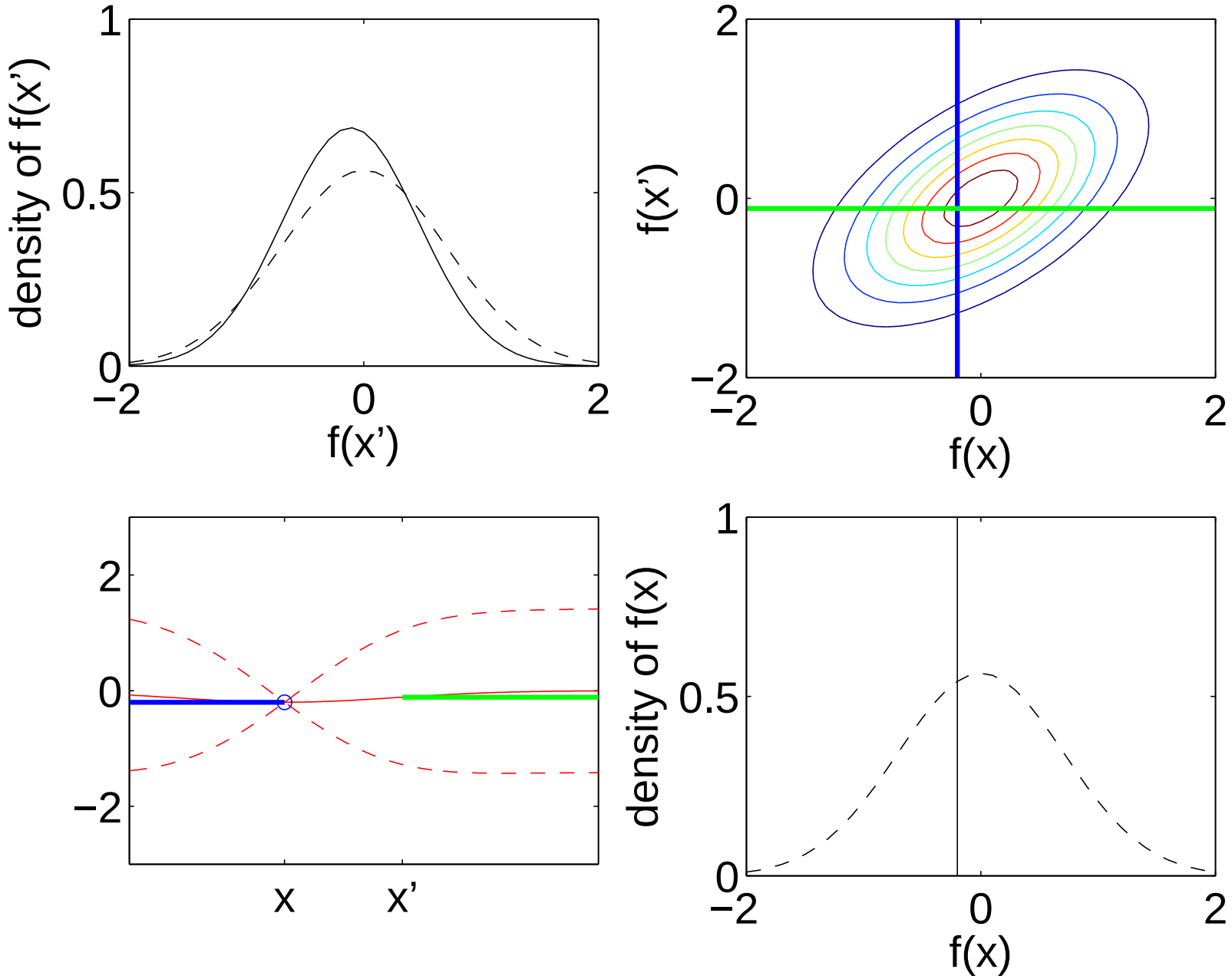




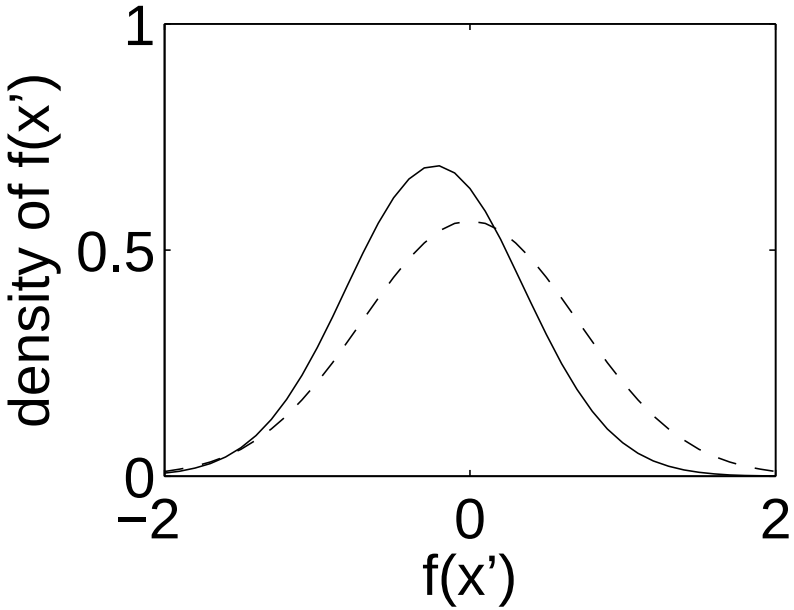


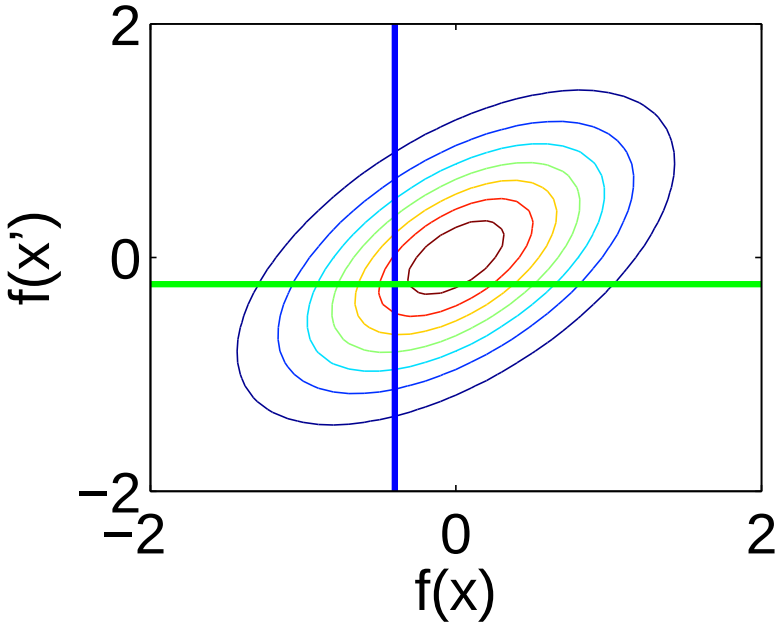


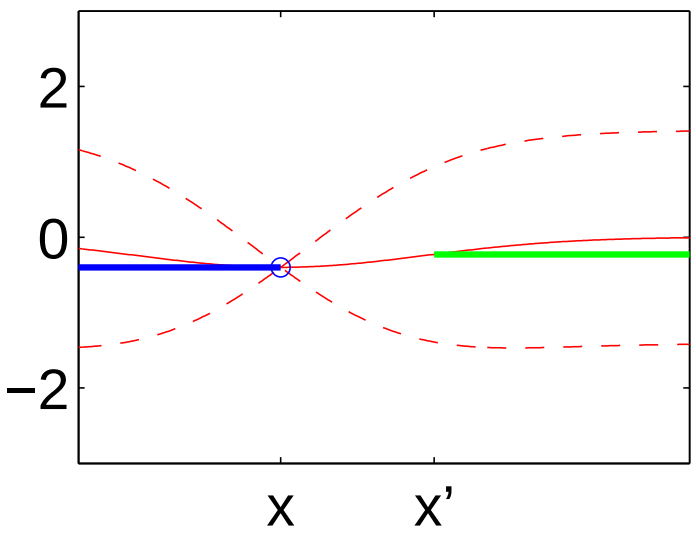
Gaussian Process Regression: Two Points

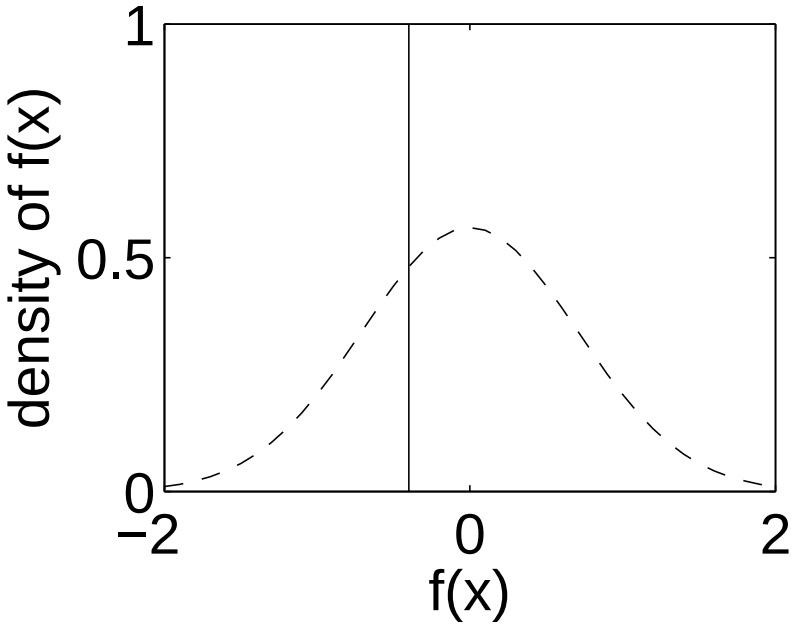


Gaussian Process Regression: Two Points

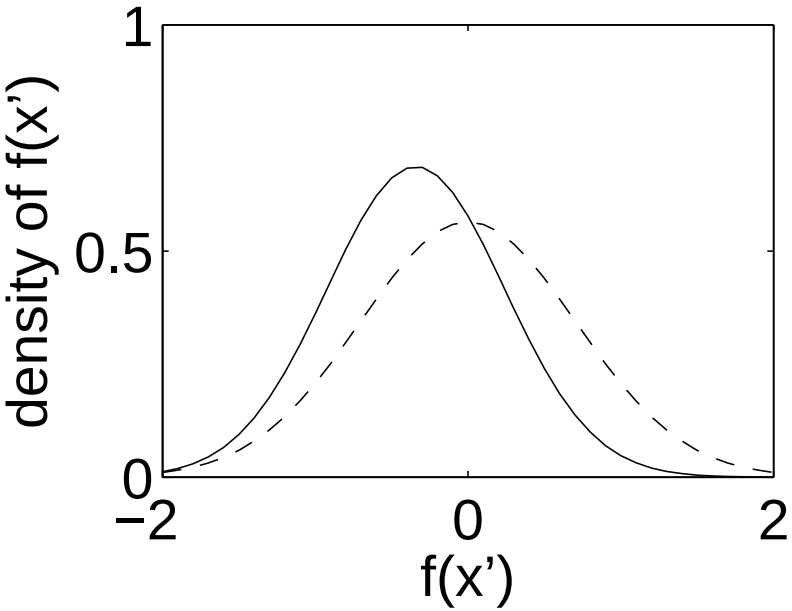


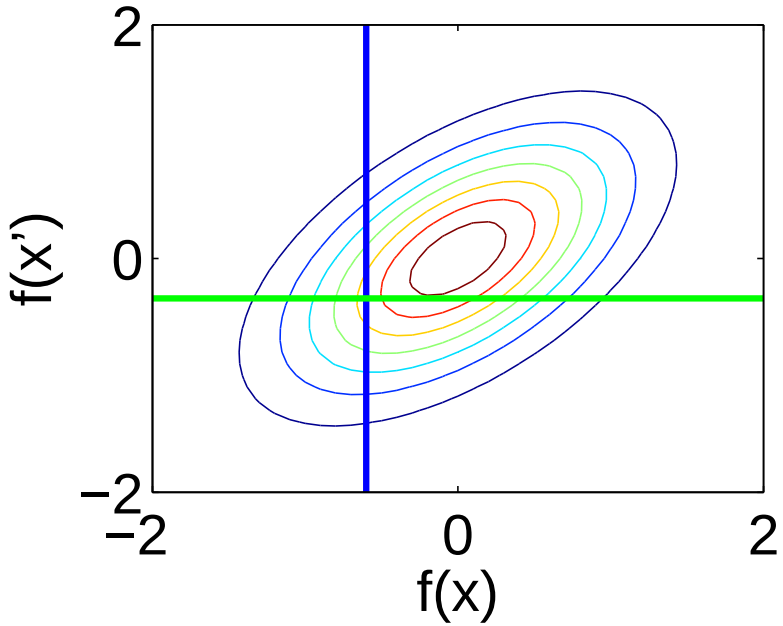


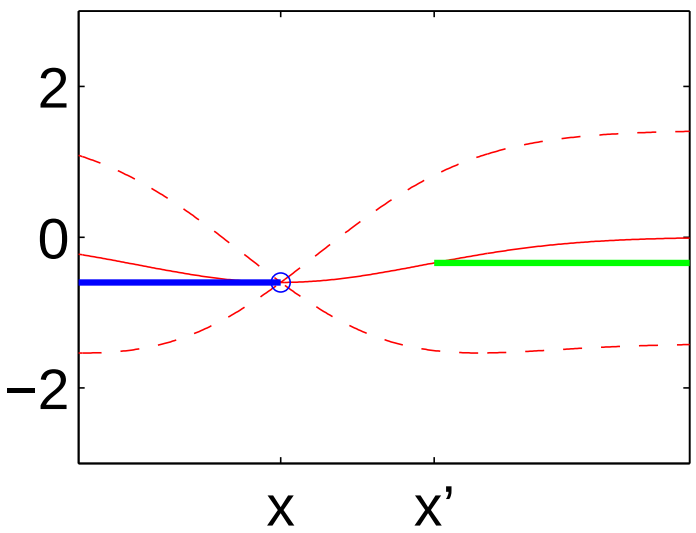


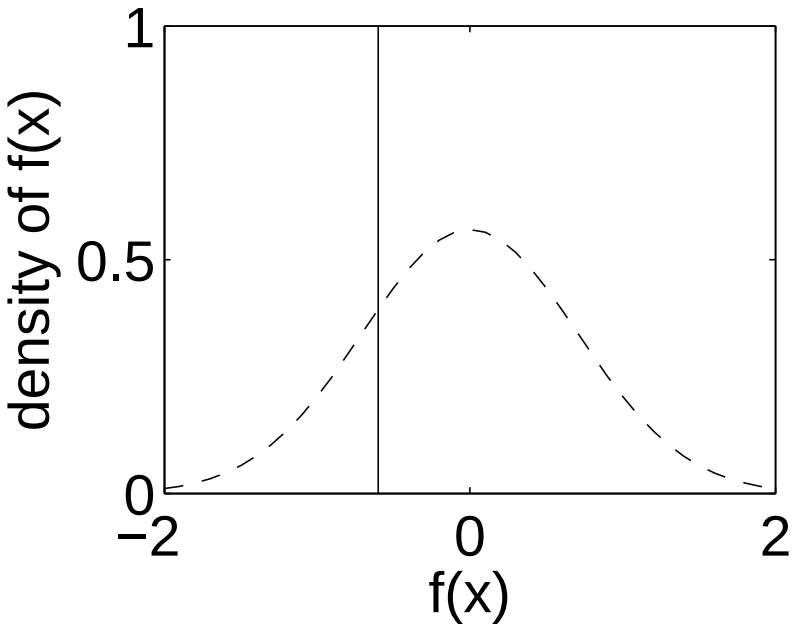


Gaussian Process Regression: Two Points

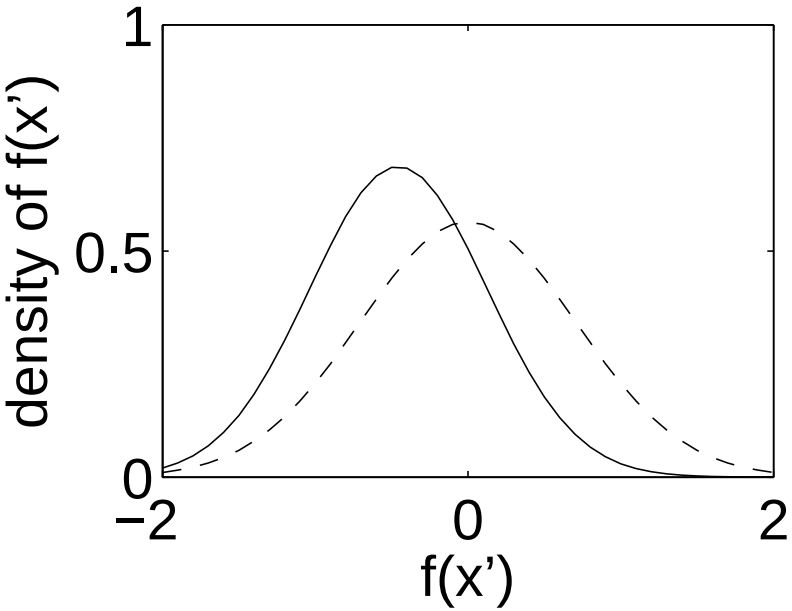


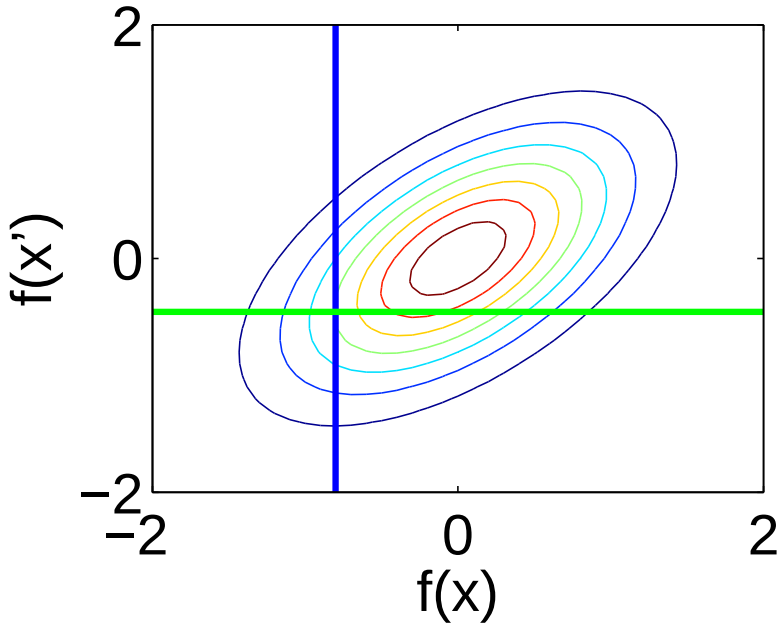


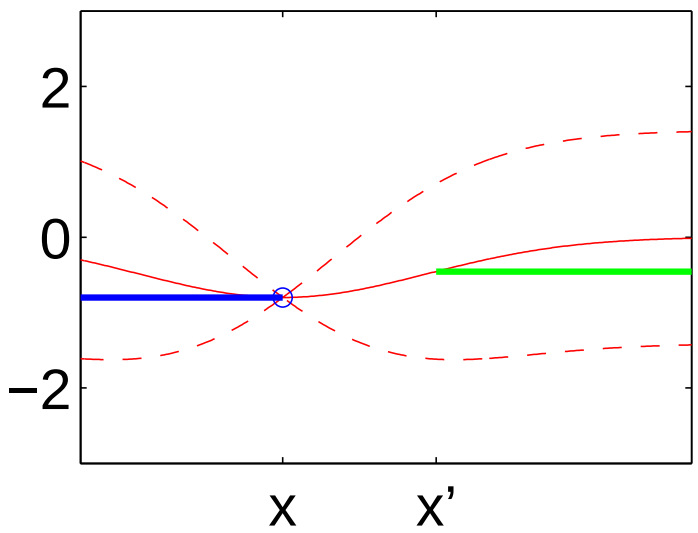


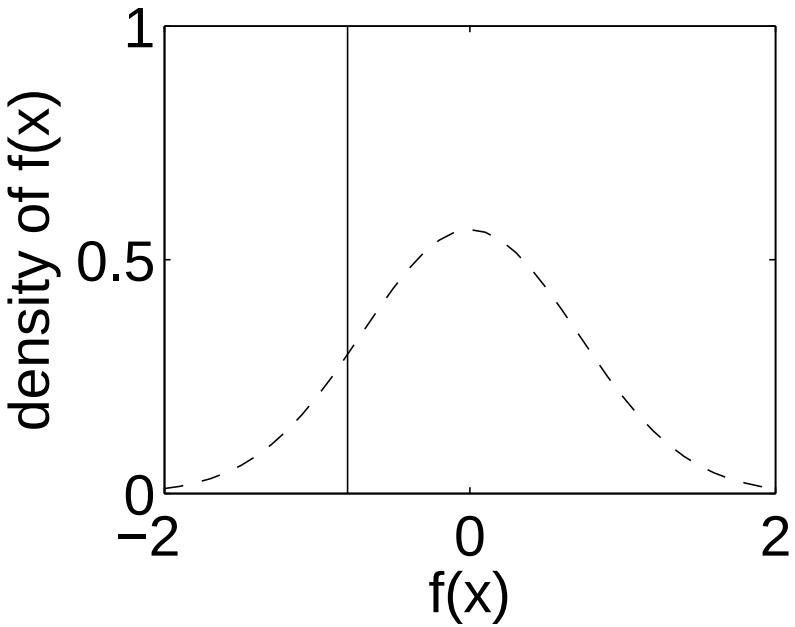


Gaussian Process Regression: Two Points



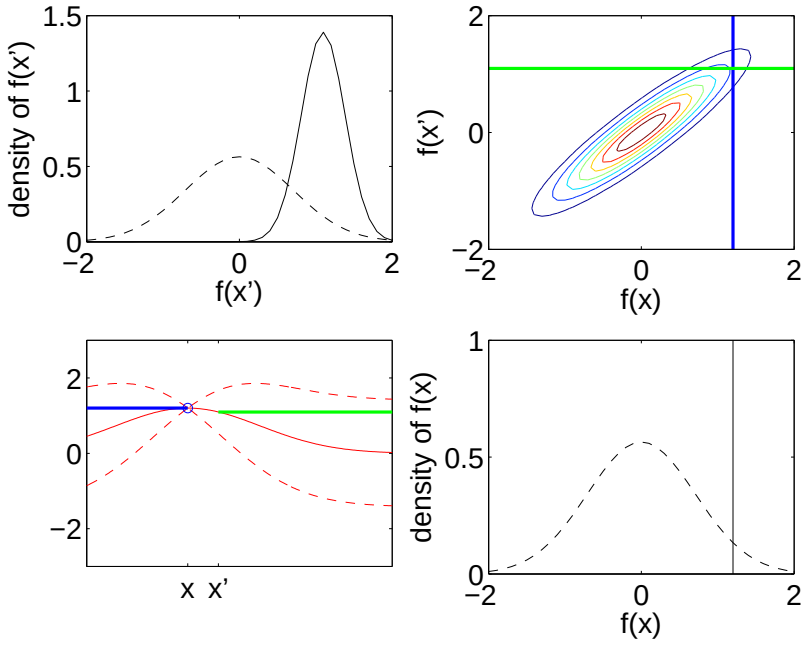






Nearby Points Have Stronger Correlation

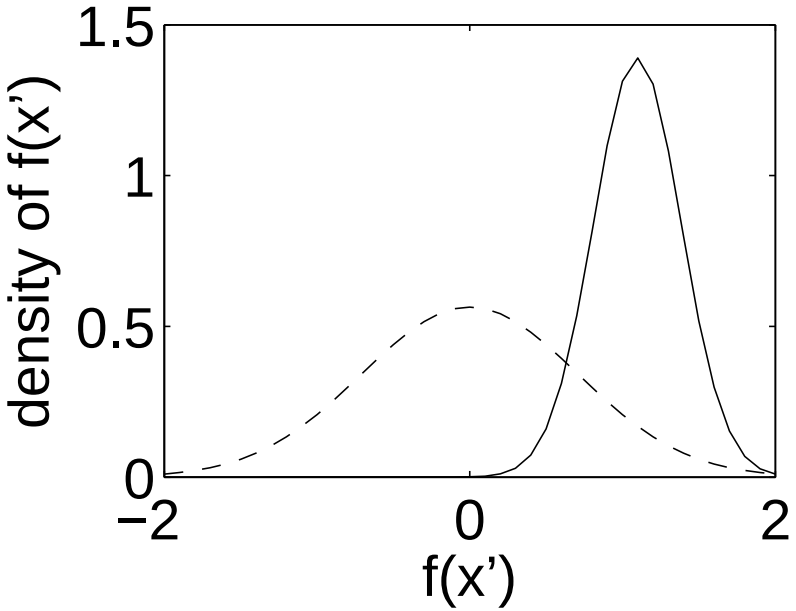
- The closer x and x' are in the feasible domain, the stronger the correlation under our belief between $f(x)$ and $f(x')$.

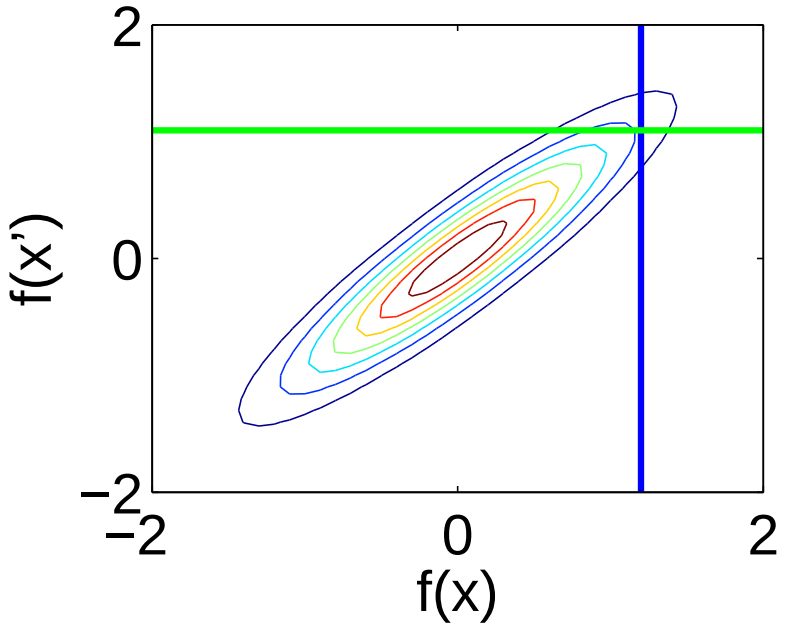


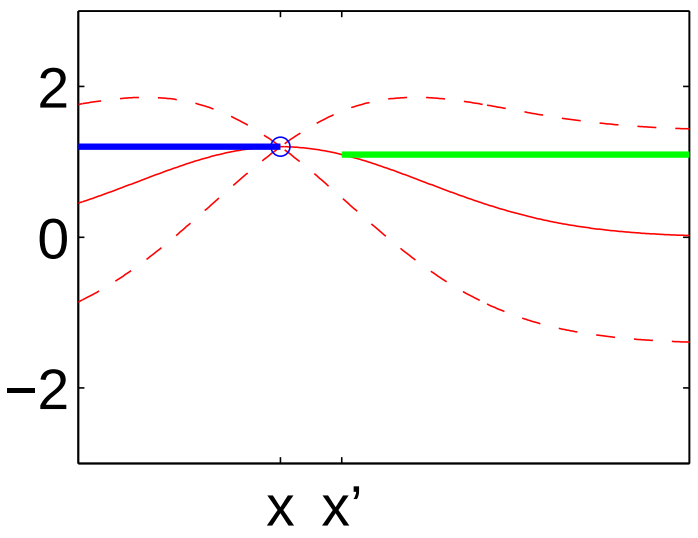
- This should be enforced by our choice of $\Sigma_0(\cdot,\cdot)$. A common choice is the power exponential:

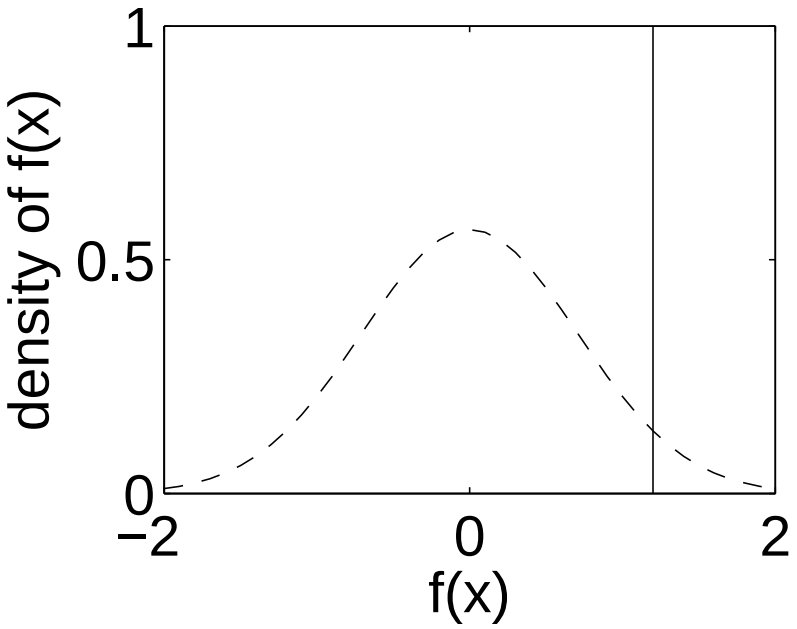
$$\Sigma_0(x,x') = \alpha_0 \exp\left(-\alpha_1 ||x - x'||^2\right)$$

Gaussian Process Regression: Two Close Points

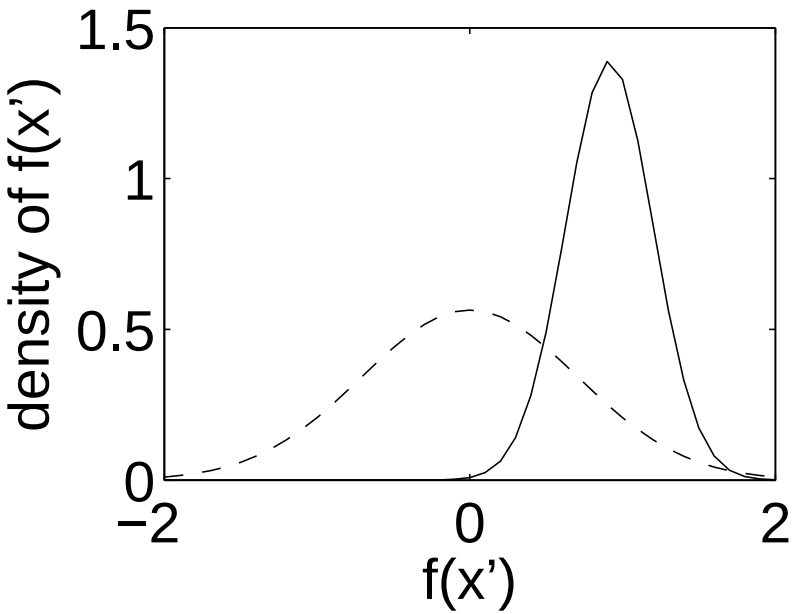


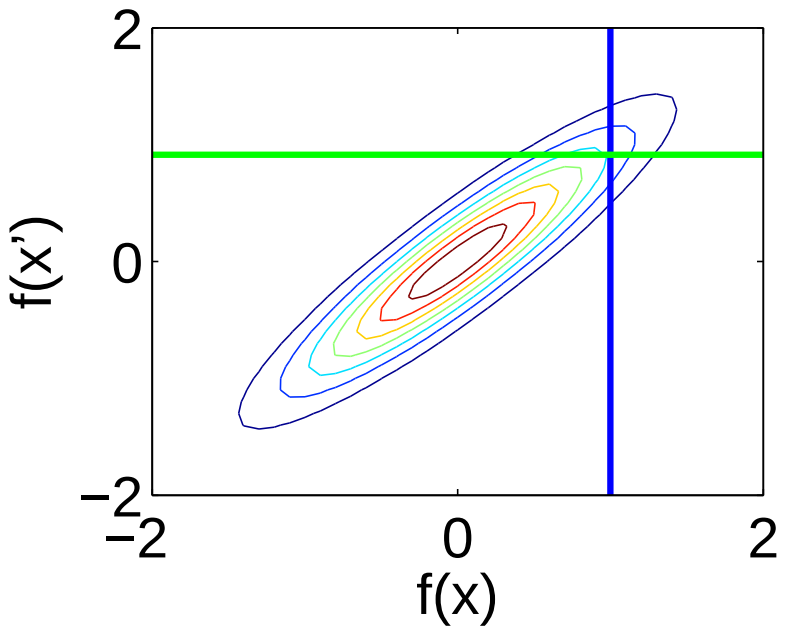


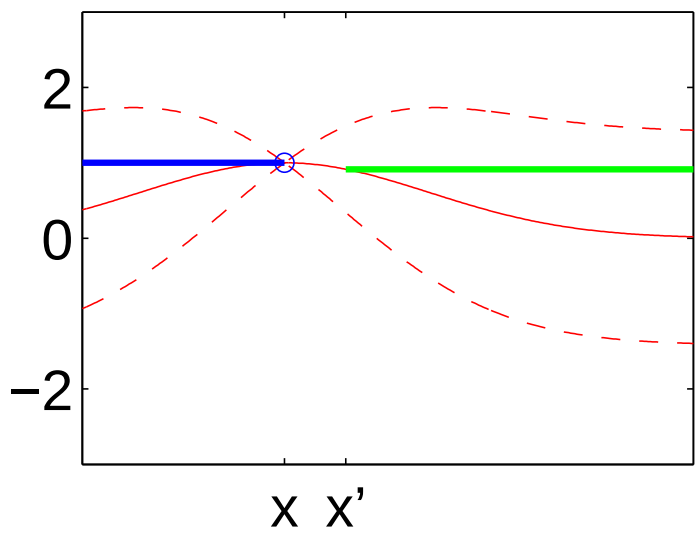


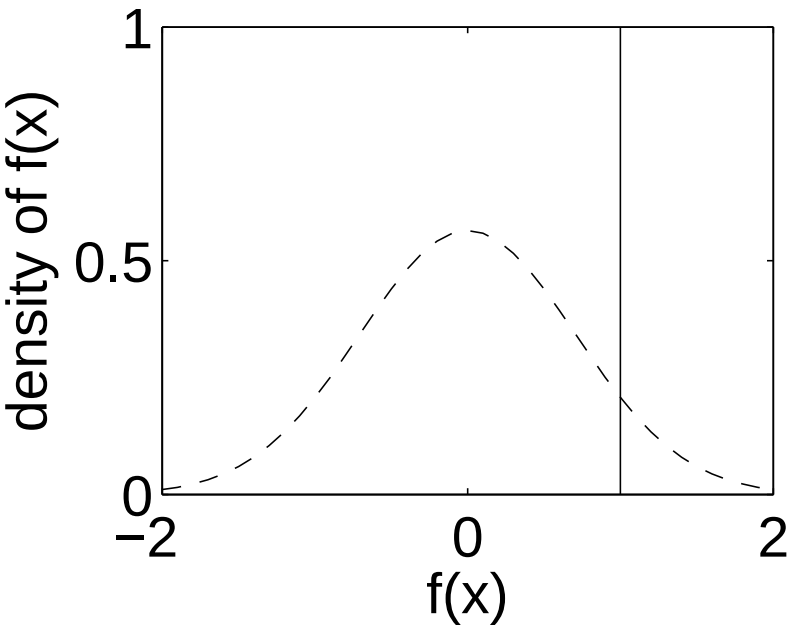


Gaussian Process Regression: Two Close Points

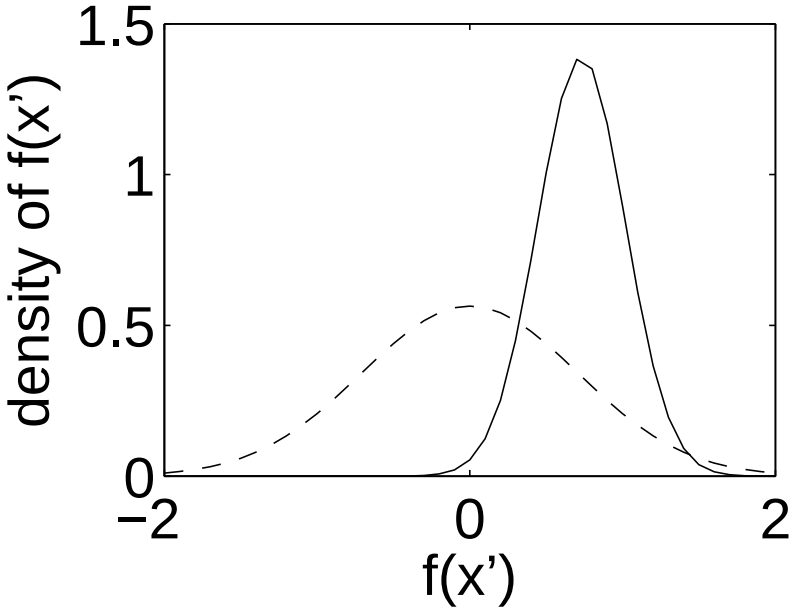


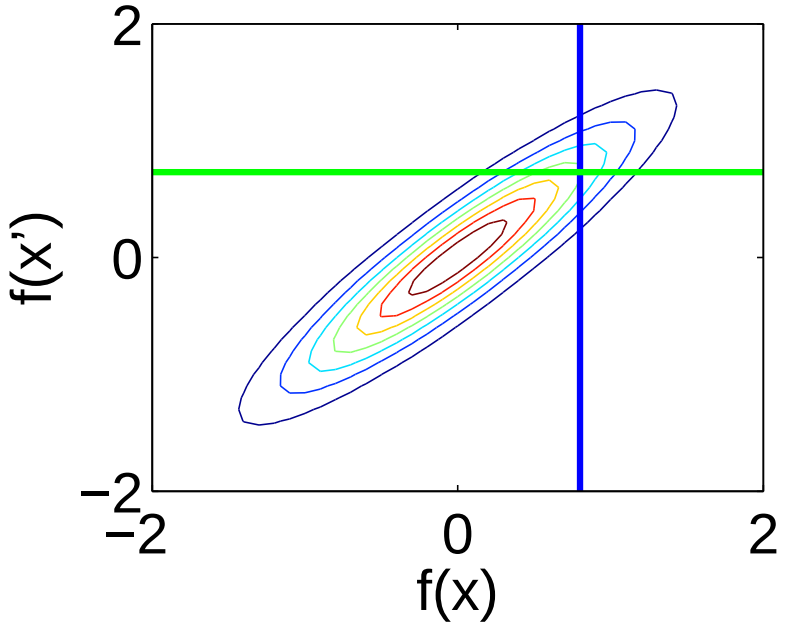


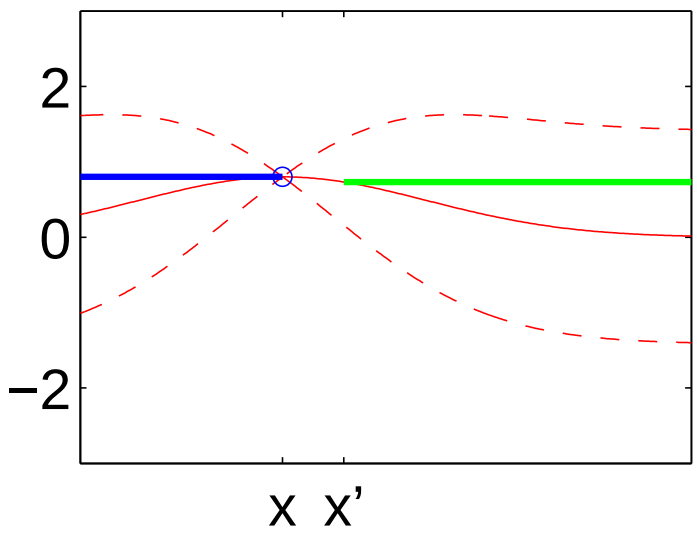


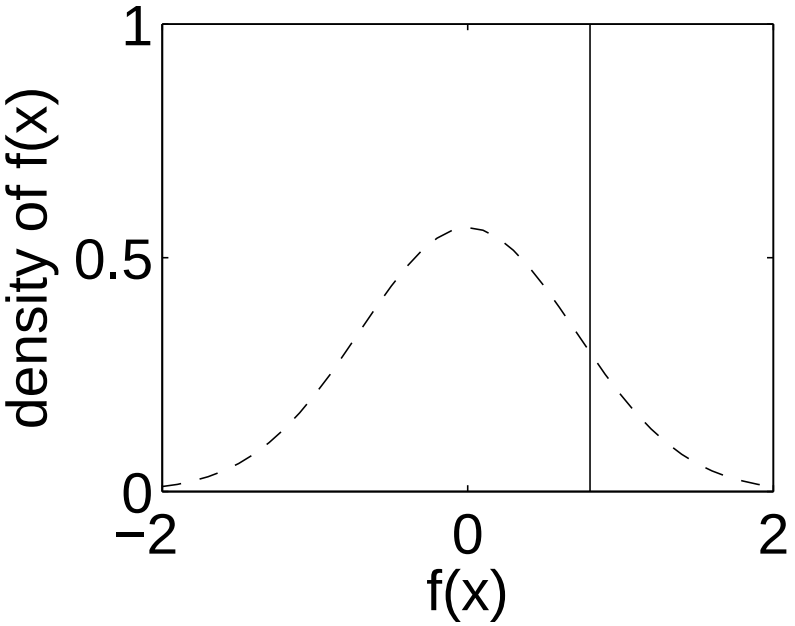


Gaussian Process Regression: Two Close Points

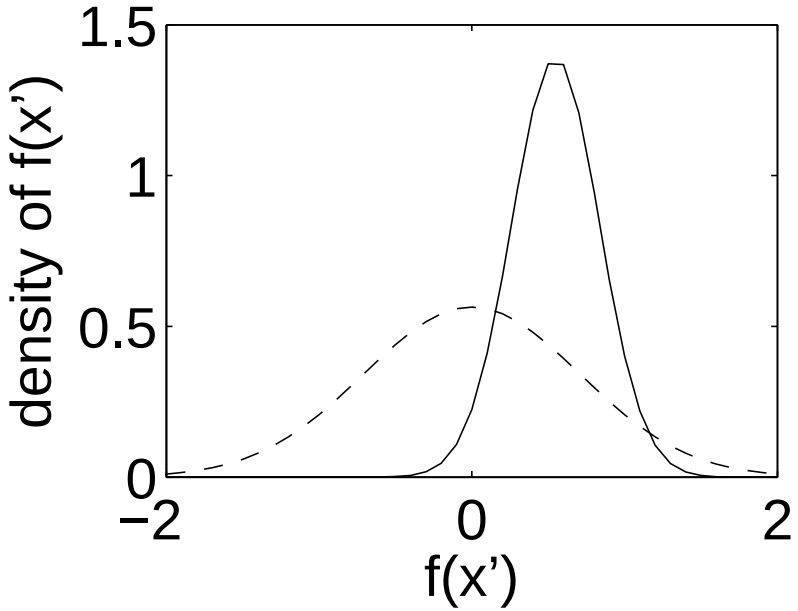


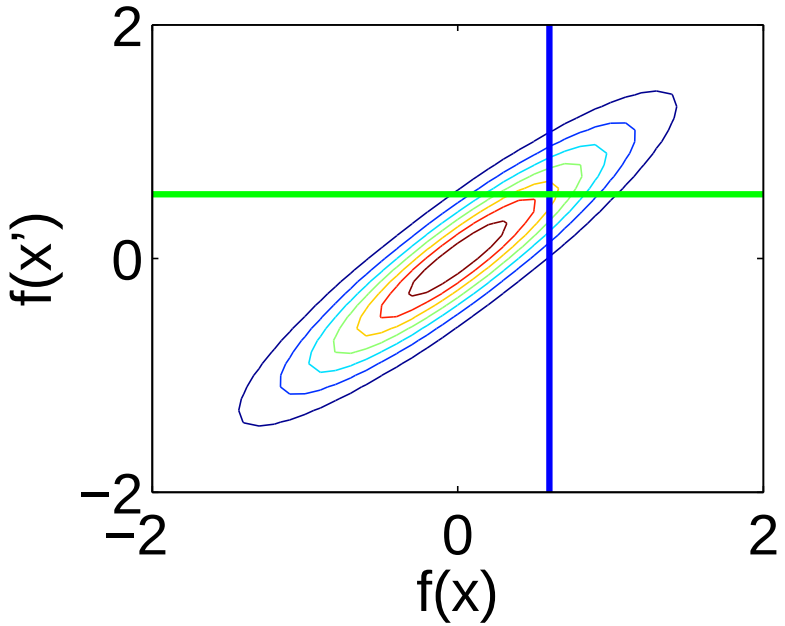


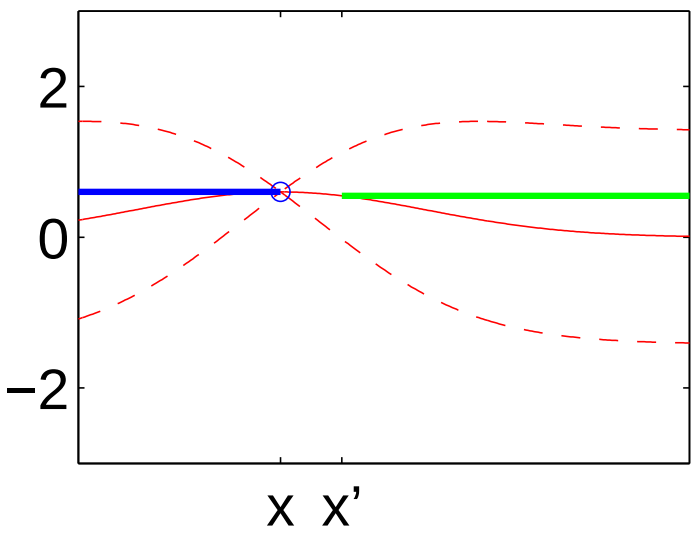


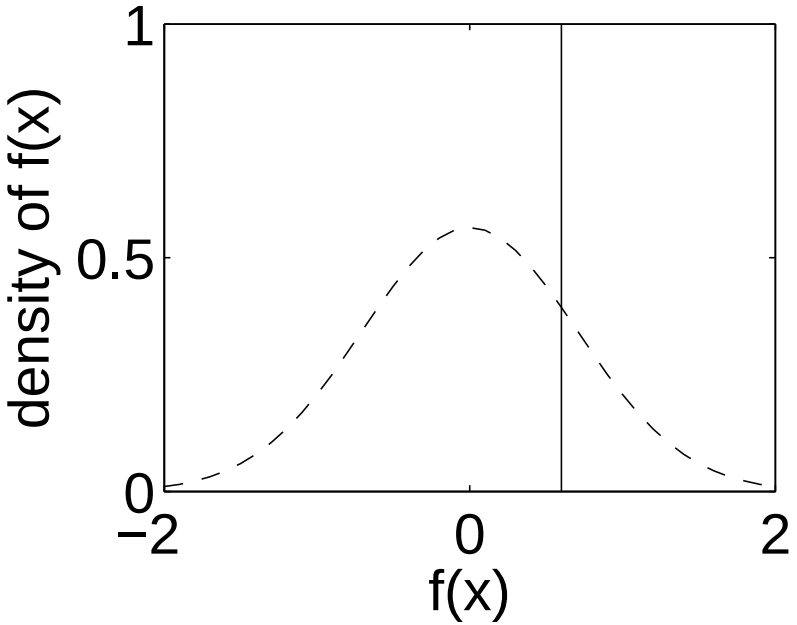


Gaussian Process Regression: Two Close Points

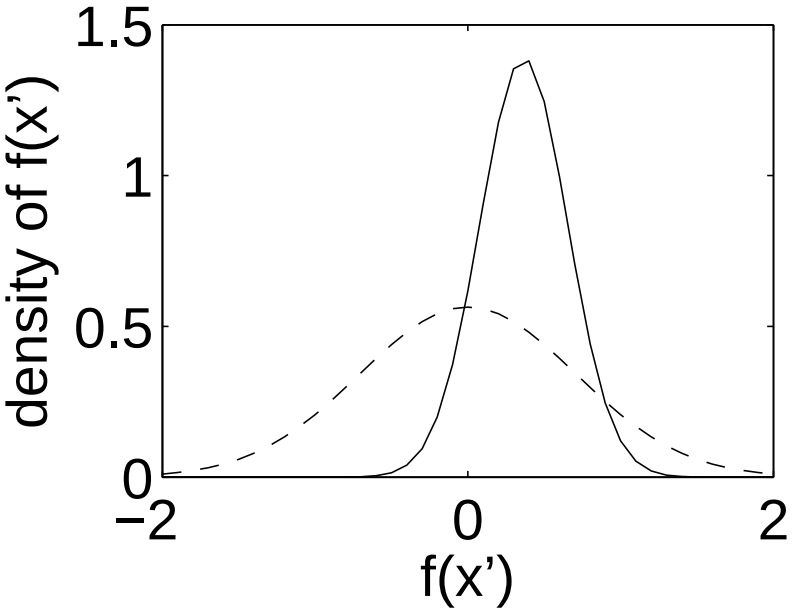


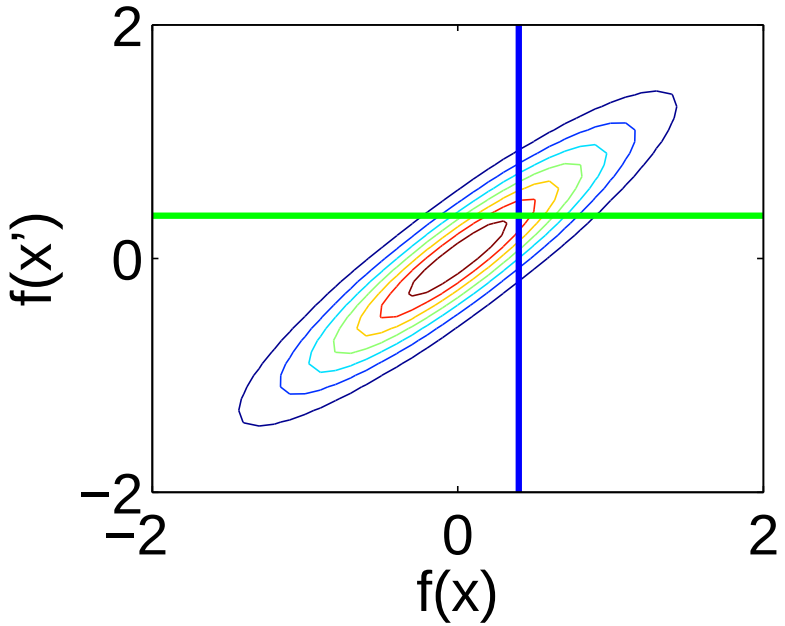


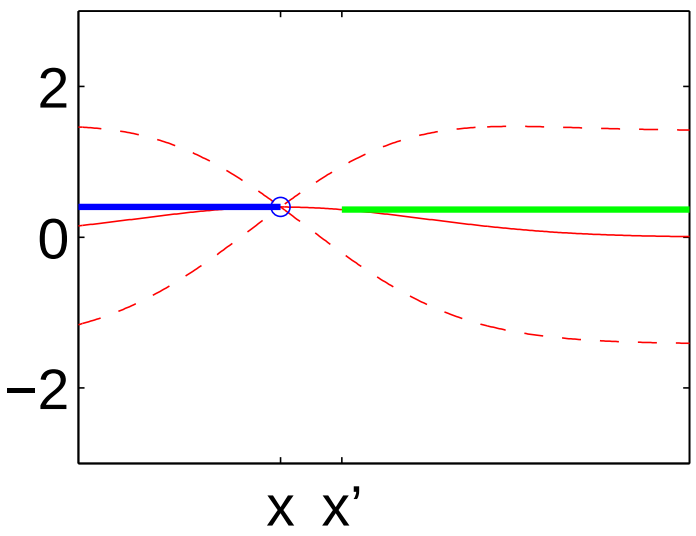


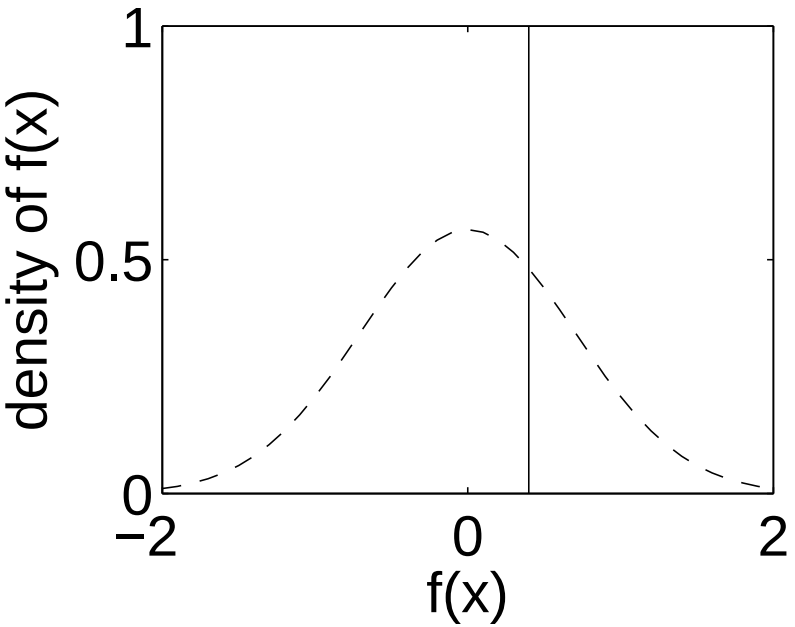


Gaussian Process Regression: Two Close Points

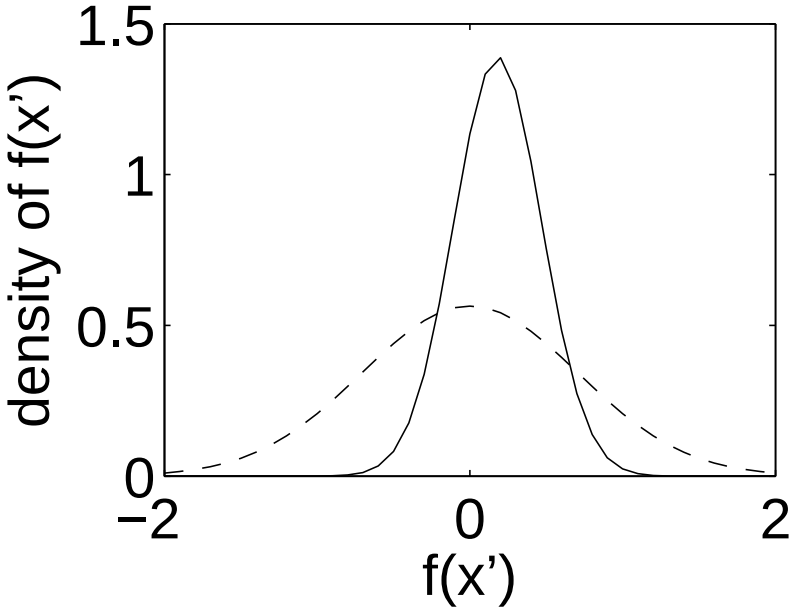


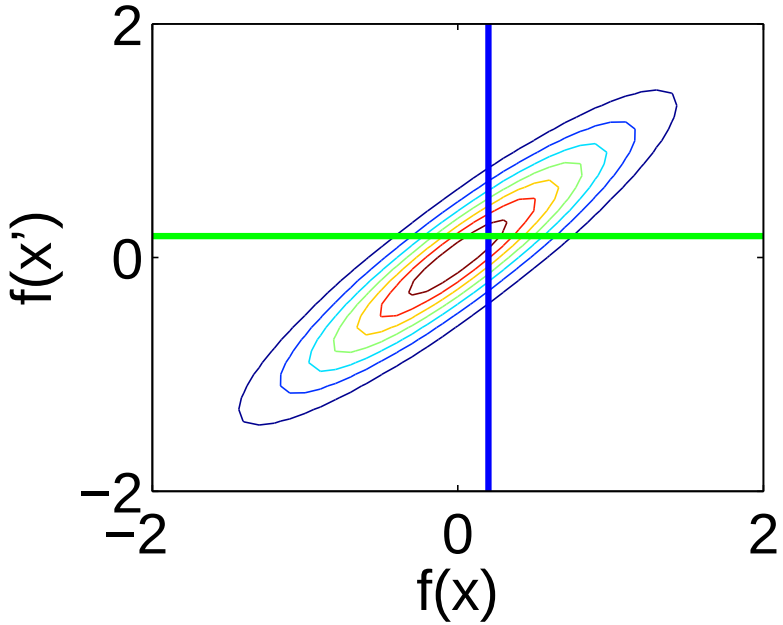


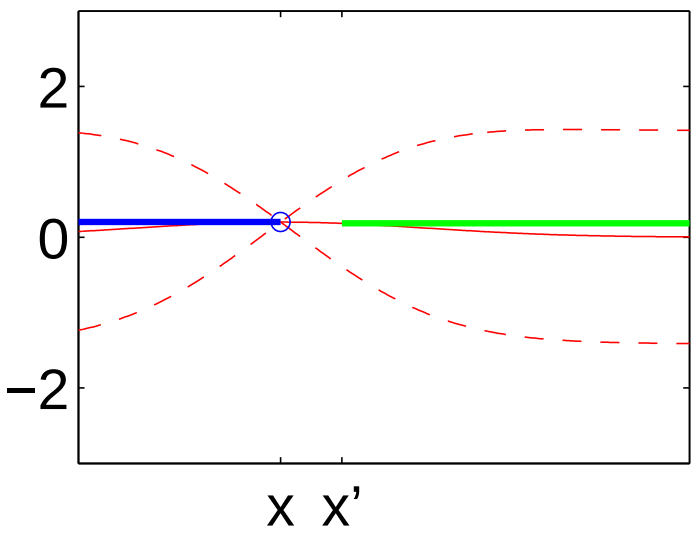


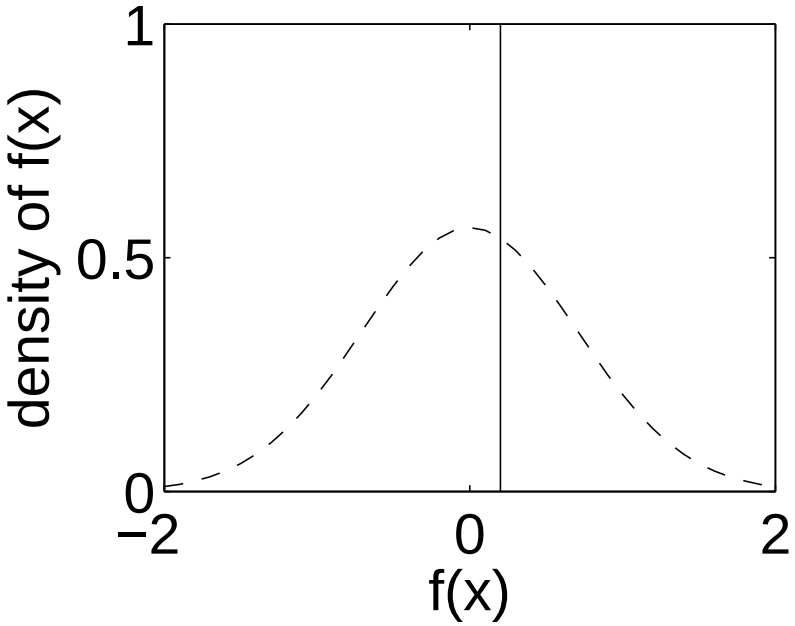


Gaussian Process Regression: Two Close Points

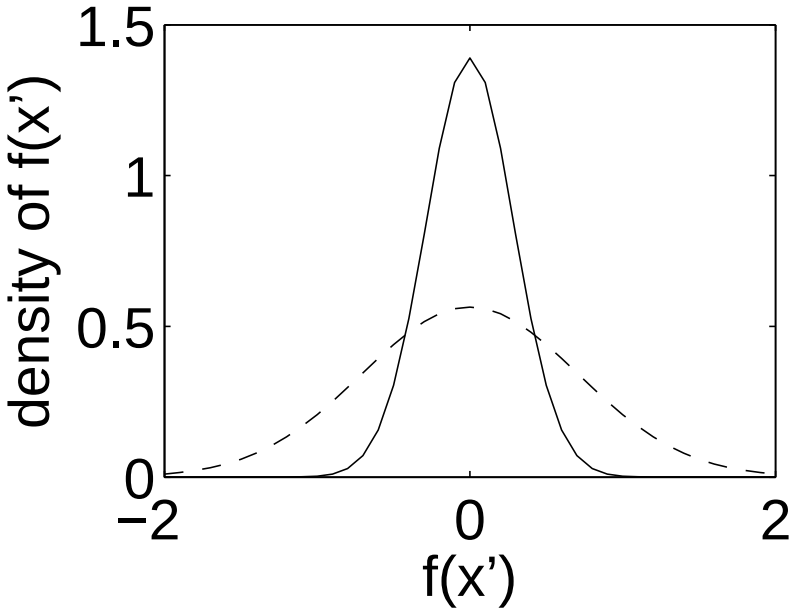


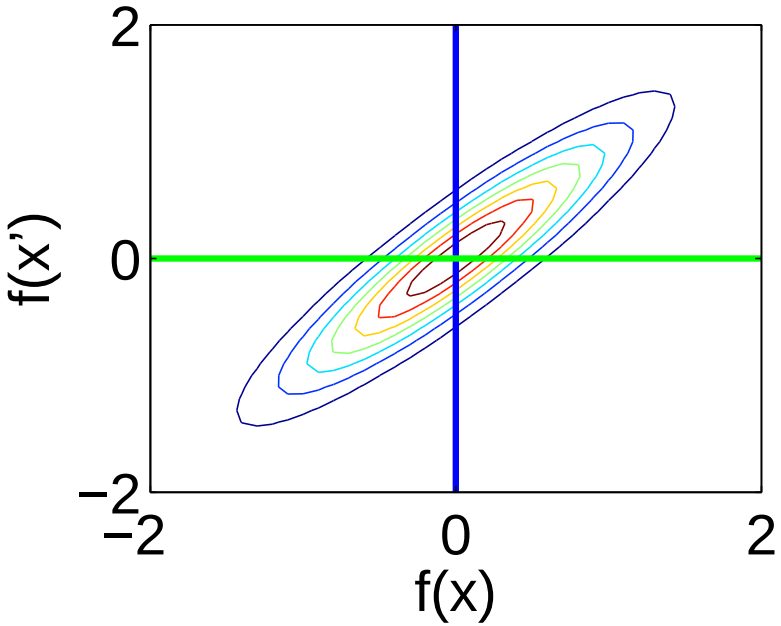


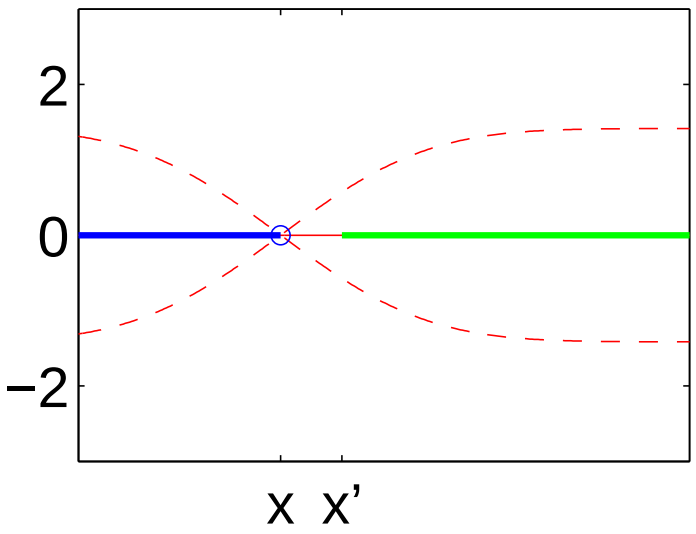


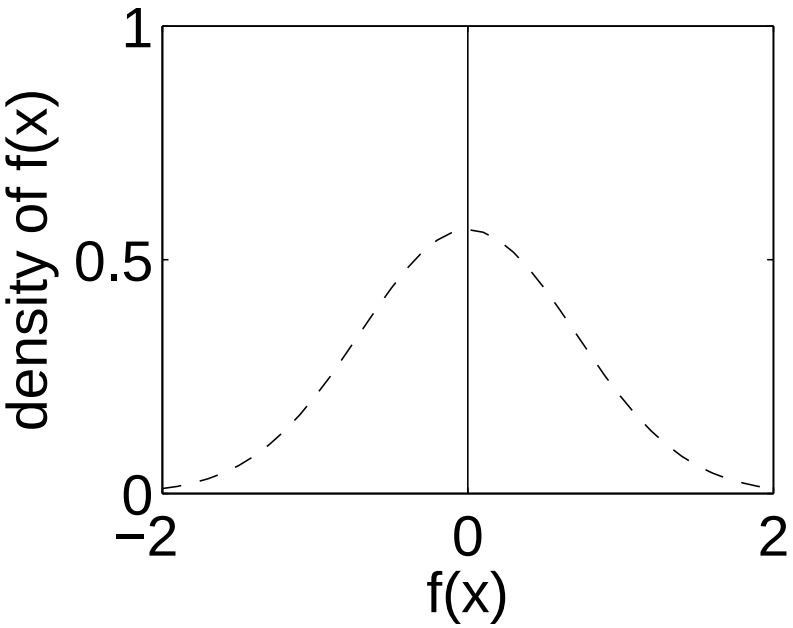


Gaussian Process Regression: Two Close Points

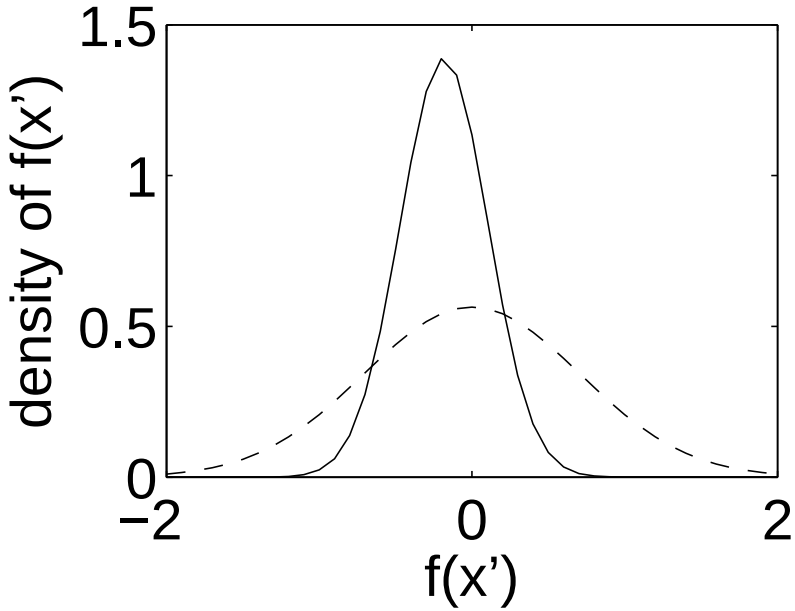


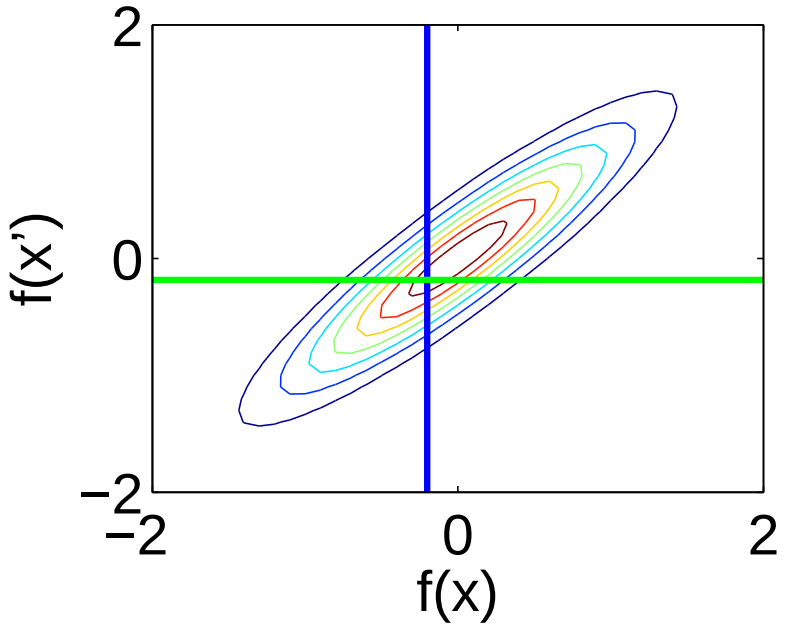


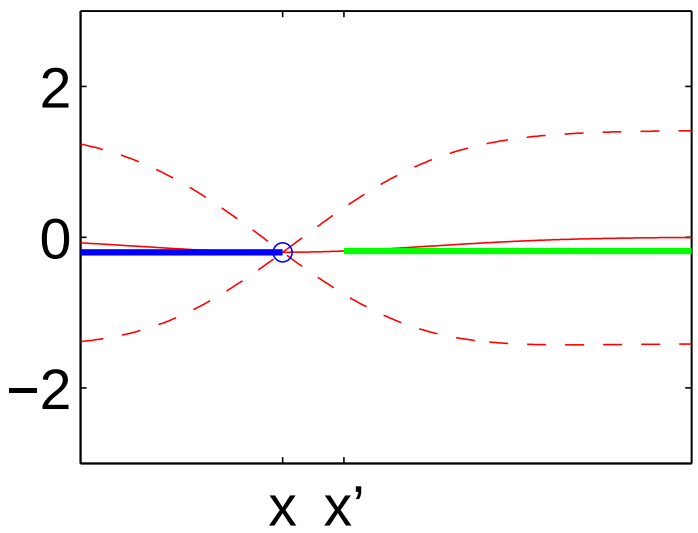


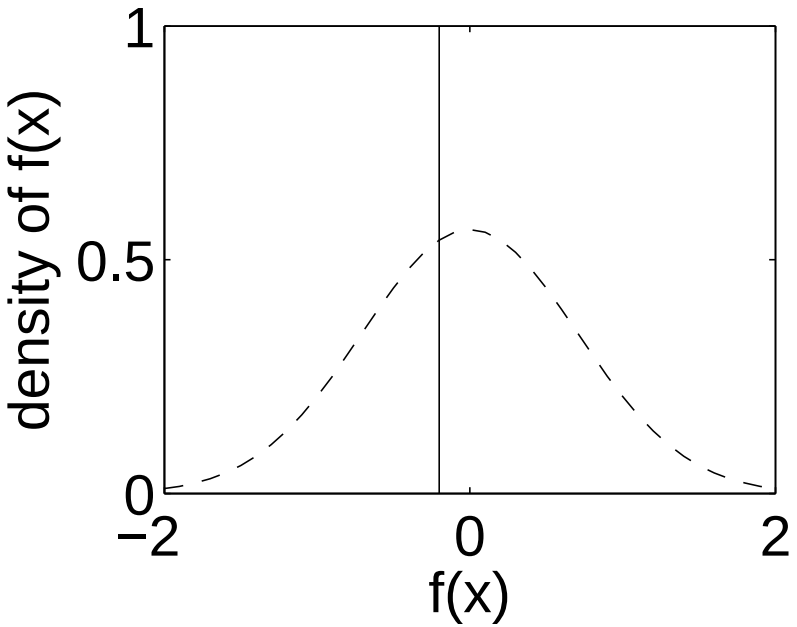


Gaussian Process Regression: Two Close Points

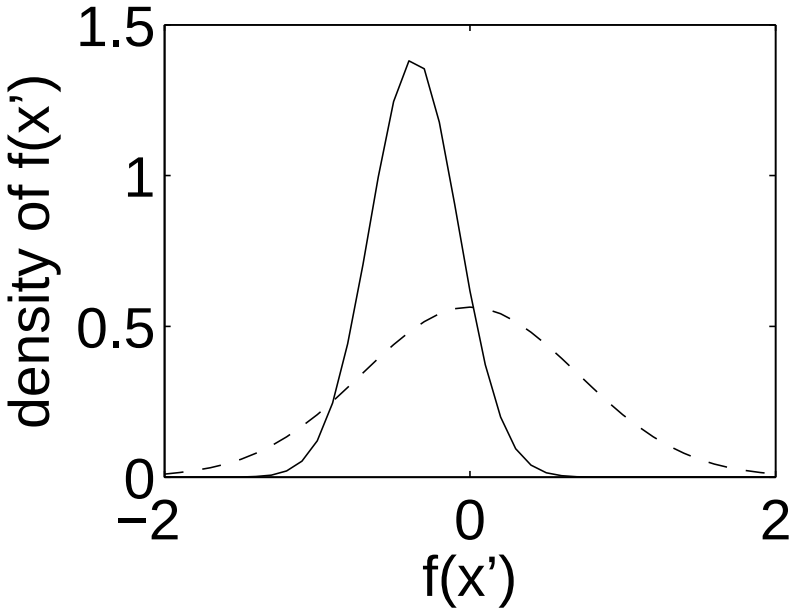


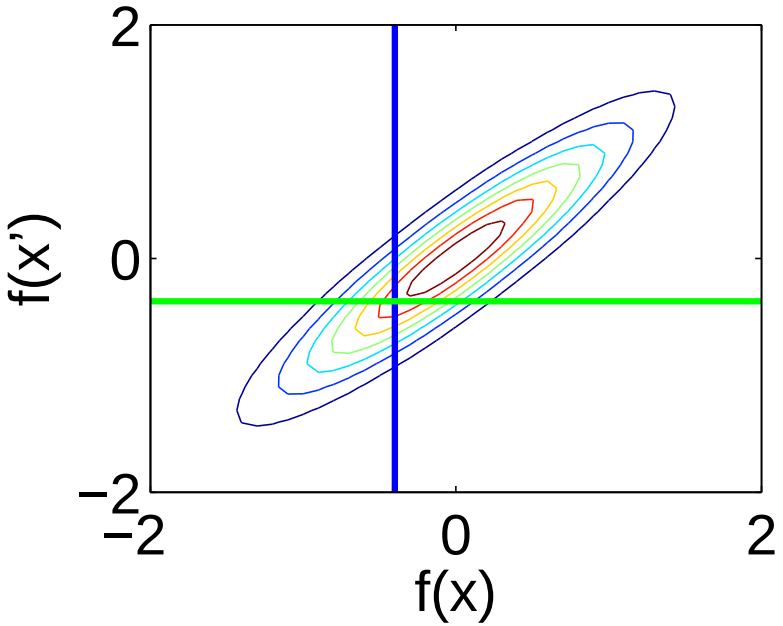


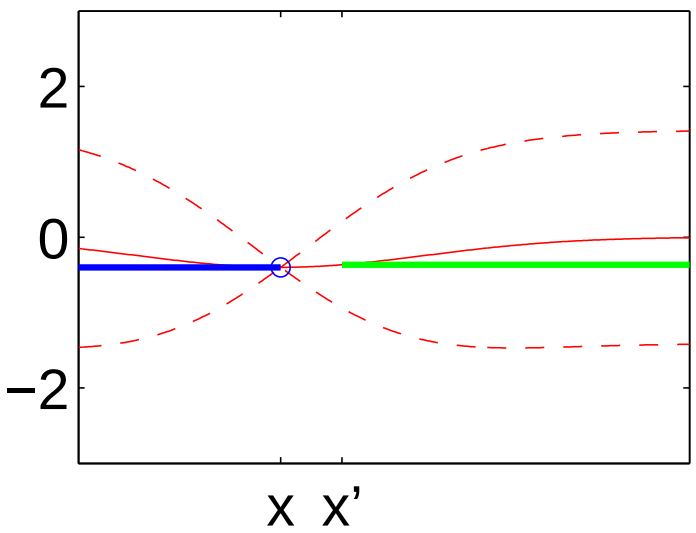


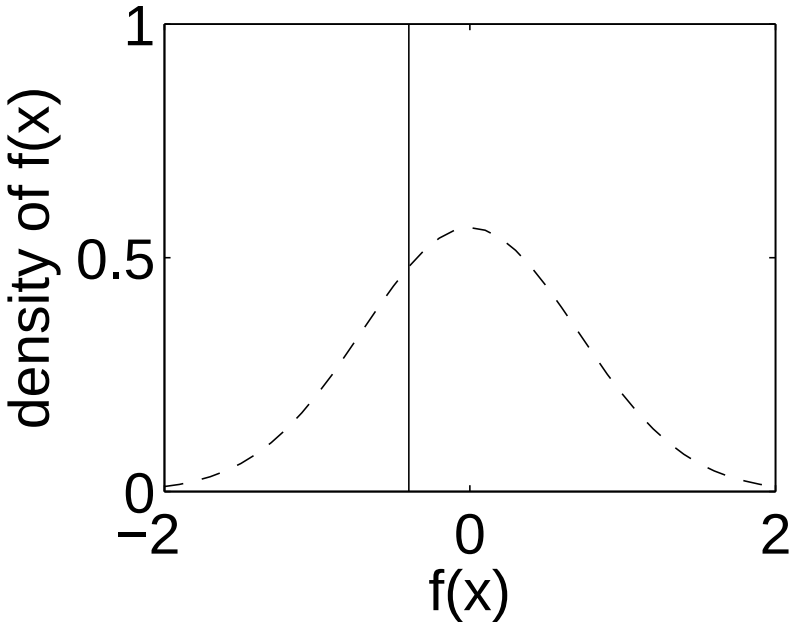


Gaussian Process Regression: Two Close Points

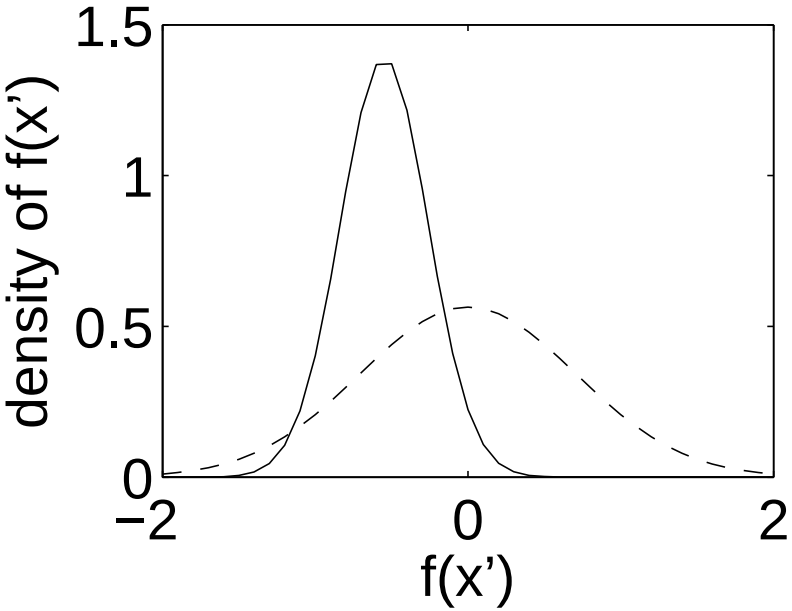


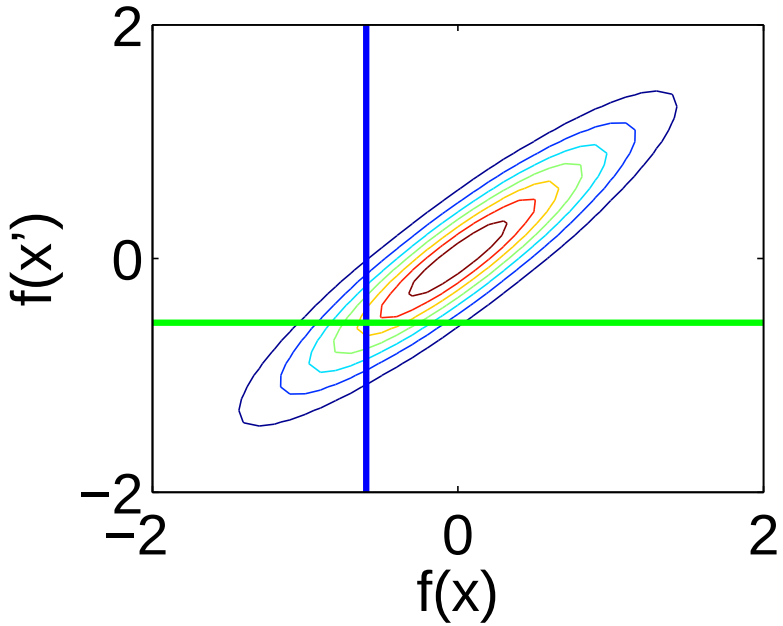


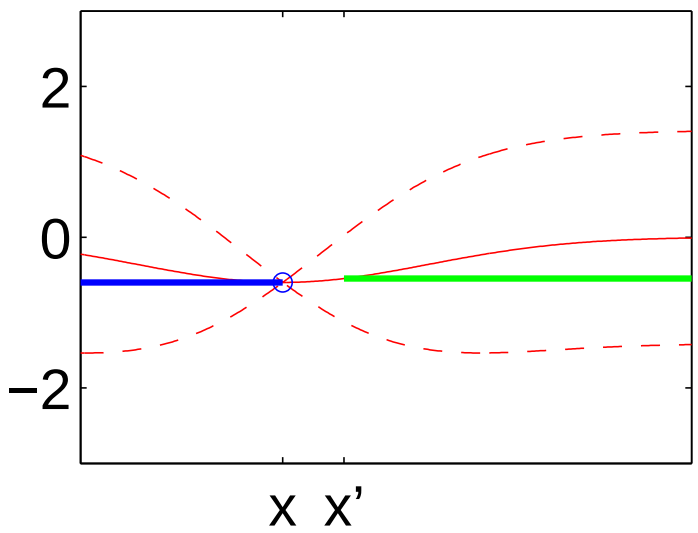


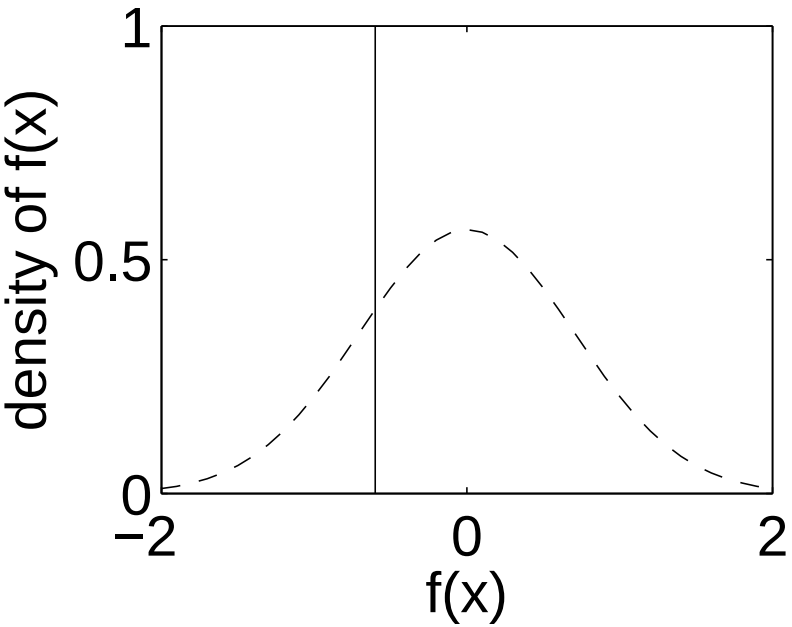


Gaussian Process Regression: Two Close Points

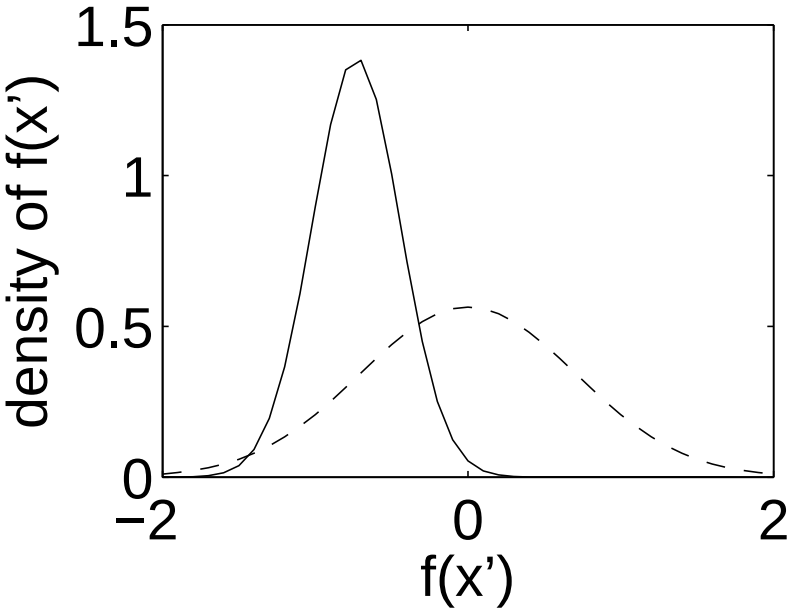


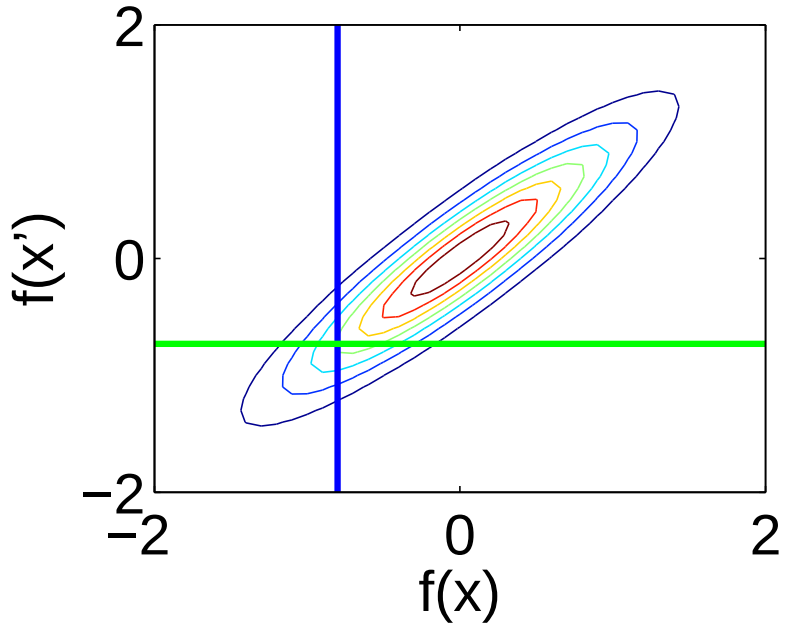


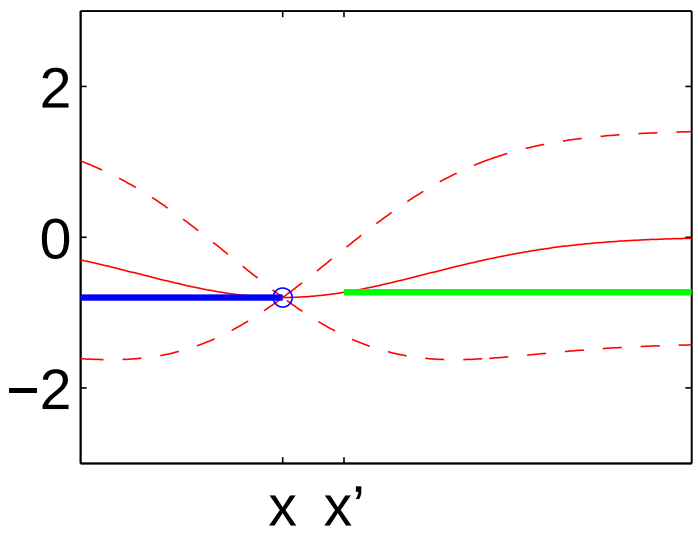


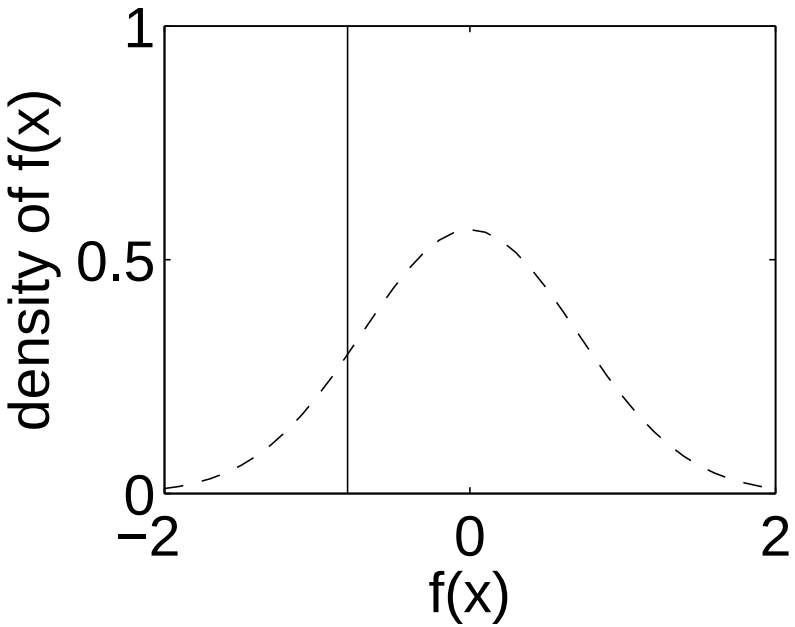


Gaussian Process Regression: Two Close Points









GP Regression: Formal Definition

A GP prior on an unknown function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is parameterized by a

- mean function $\mu_0(\cdot)$.
- covariance function $\Sigma_0(\cdot, \cdot)$, that must be positive semi-definite.

Definition: A prior P on a function f is a **Gaussian Process (GP) prior** with mean function μ_0 and covariance function Σ_0 if:
For any given set of points $x_1, \ldots, x_k$, under P,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_0(x_1) \\ \vdots \\ \mu_0(x_k) \end{bmatrix}, \begin{bmatrix} \Sigma_0(x_1, x_1) & \ldots & \Sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_k, x_1) & \ldots & \Sigma_0(x_k, x_k) \end{bmatrix} \right)$$

The Posterior Can be Computed Analytically

Suppose we have observed

$$f(\vec{x}) = [f(x_1), \ldots, f(x_n)].$$

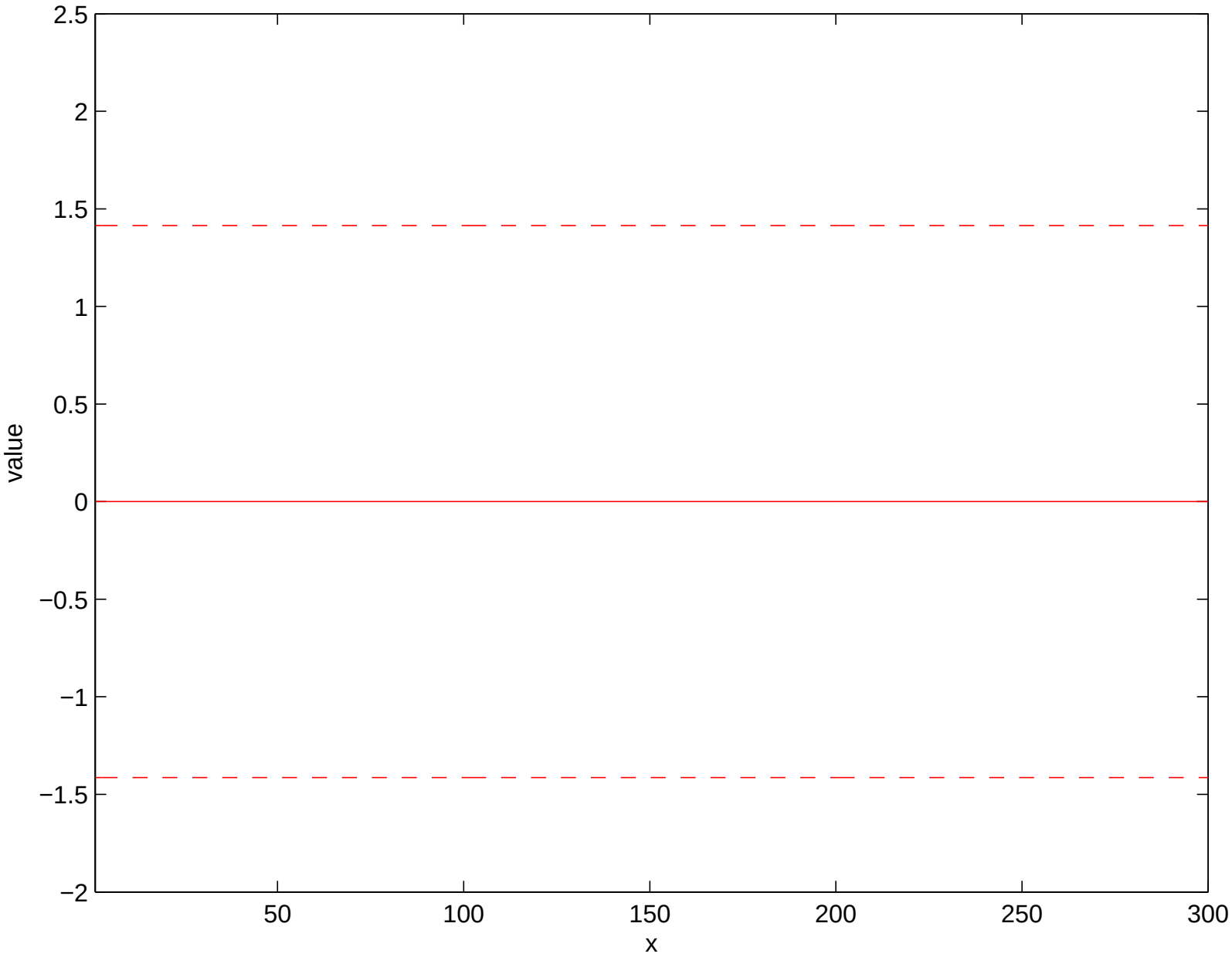
Fix any x'. The posterior on $f(x')$ is

$$f(x')|f(\vec{x}) \sim N(\mu_n(x'), \sigma_n^2(x')).$$

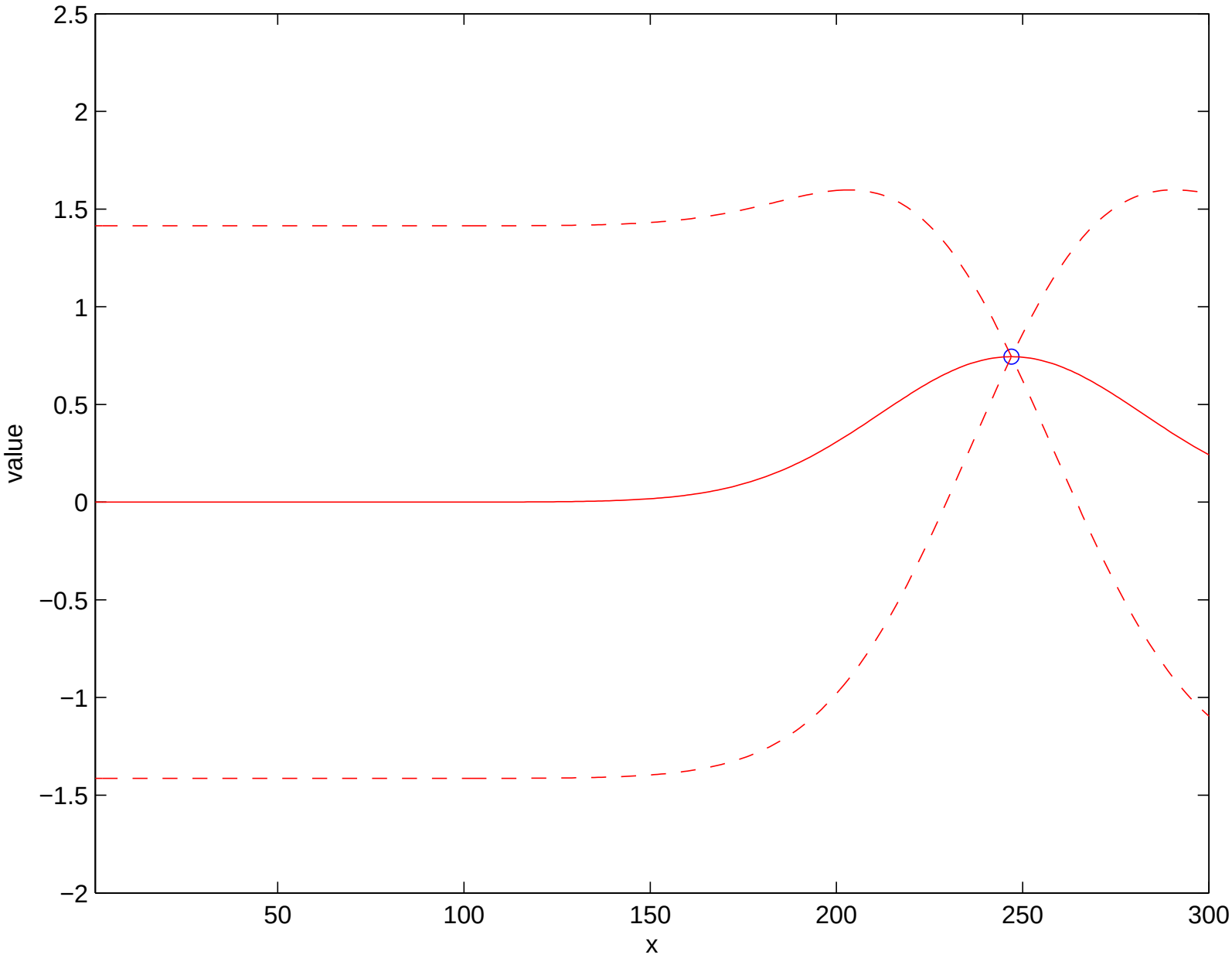
When $\mu_0(\cdot) = 0$, $\mu_n(x')$ and $\sigma_n^2(x')$ are:

$$\mu_n(x') = \Sigma_0(x', \vec{x})\Sigma_0(\vec{x}, \vec{x})^{-1} f(\vec{x}),$$
$$\sigma_n^2(x') = \Sigma_0(x', x') - \Sigma_0(x', \vec{x},)\Sigma_0(\vec{x}, \vec{x})^{-1}\Sigma_0(\vec{x}, x')$$

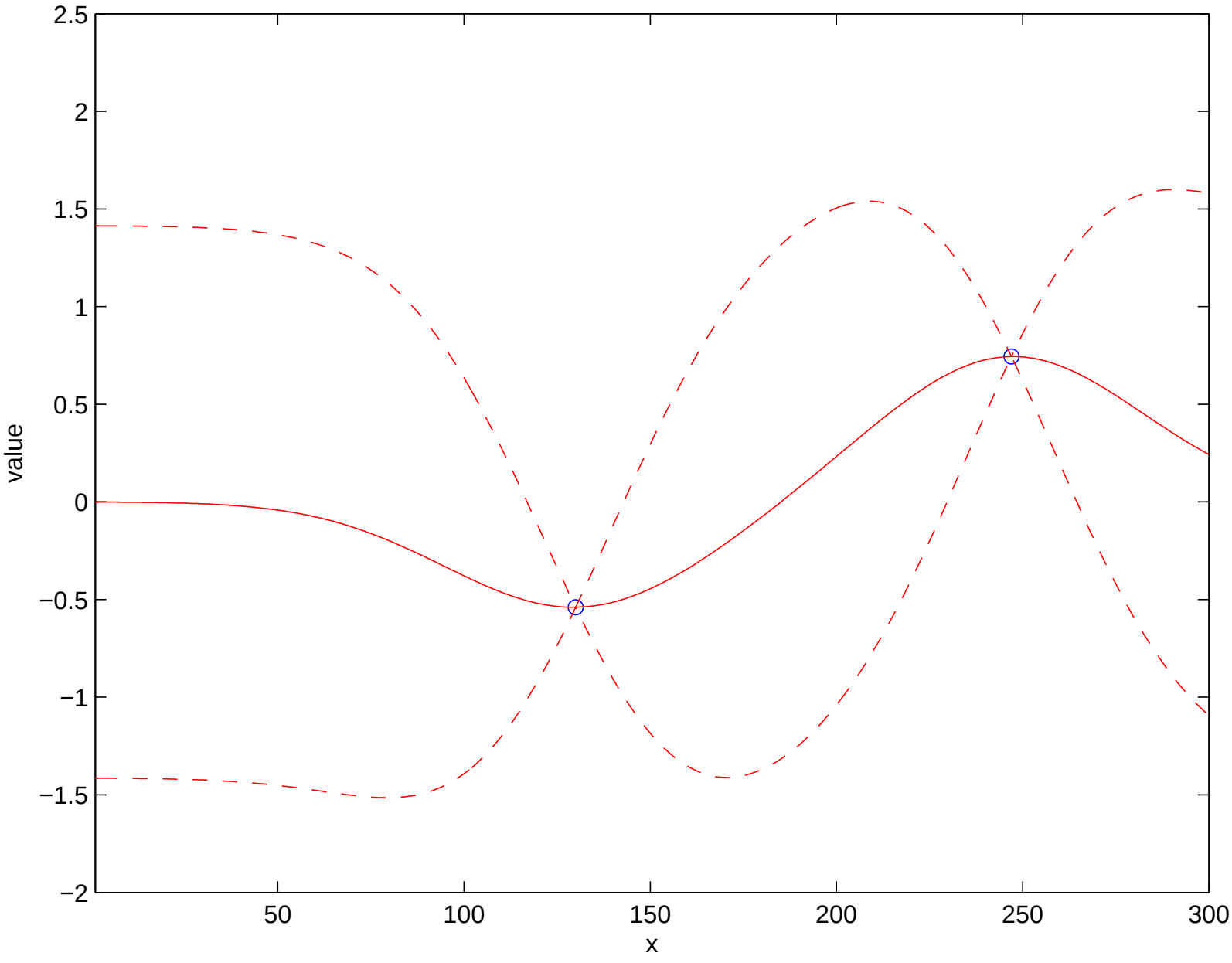
Illustrative 1D Example



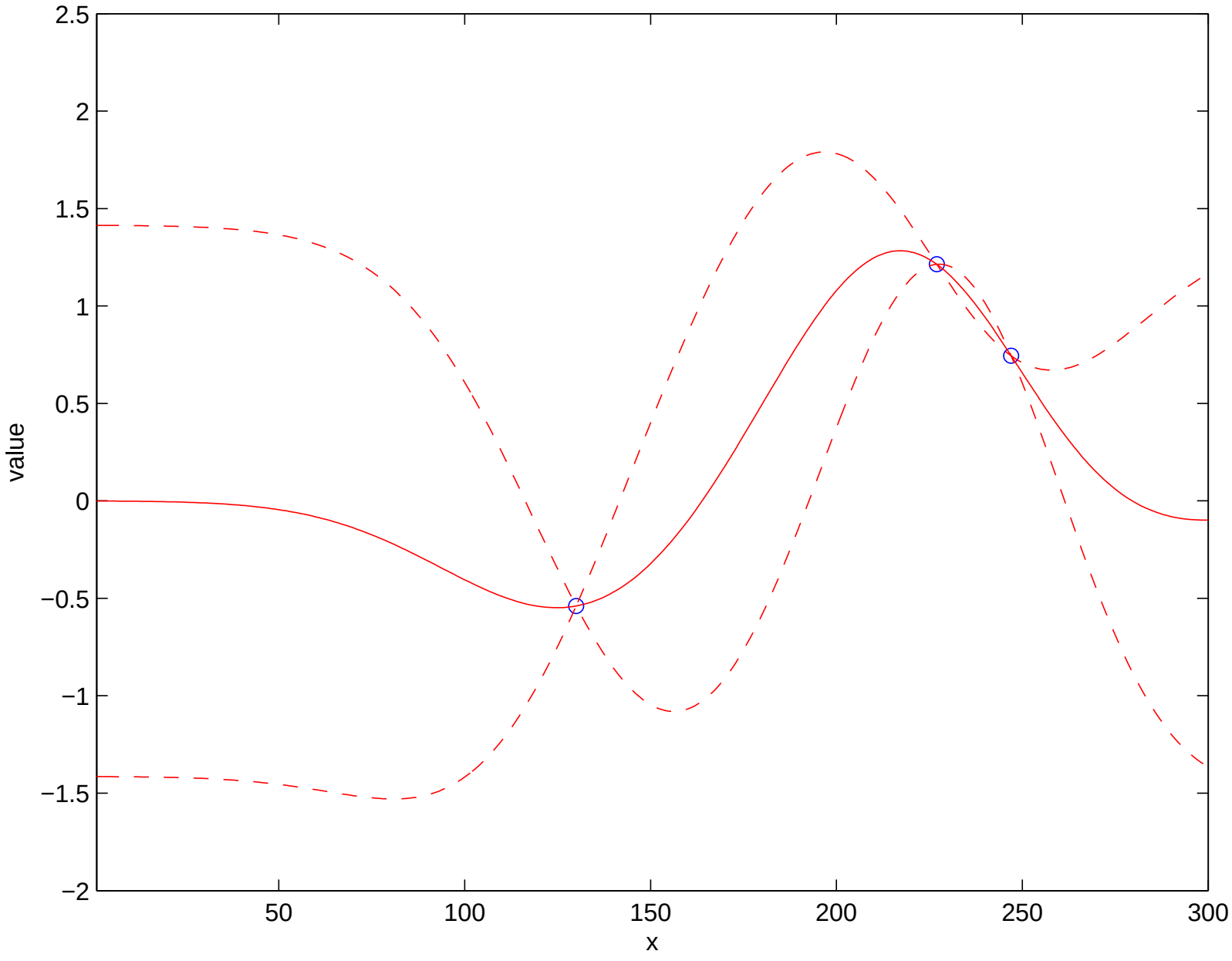
Illustrative 1D Example



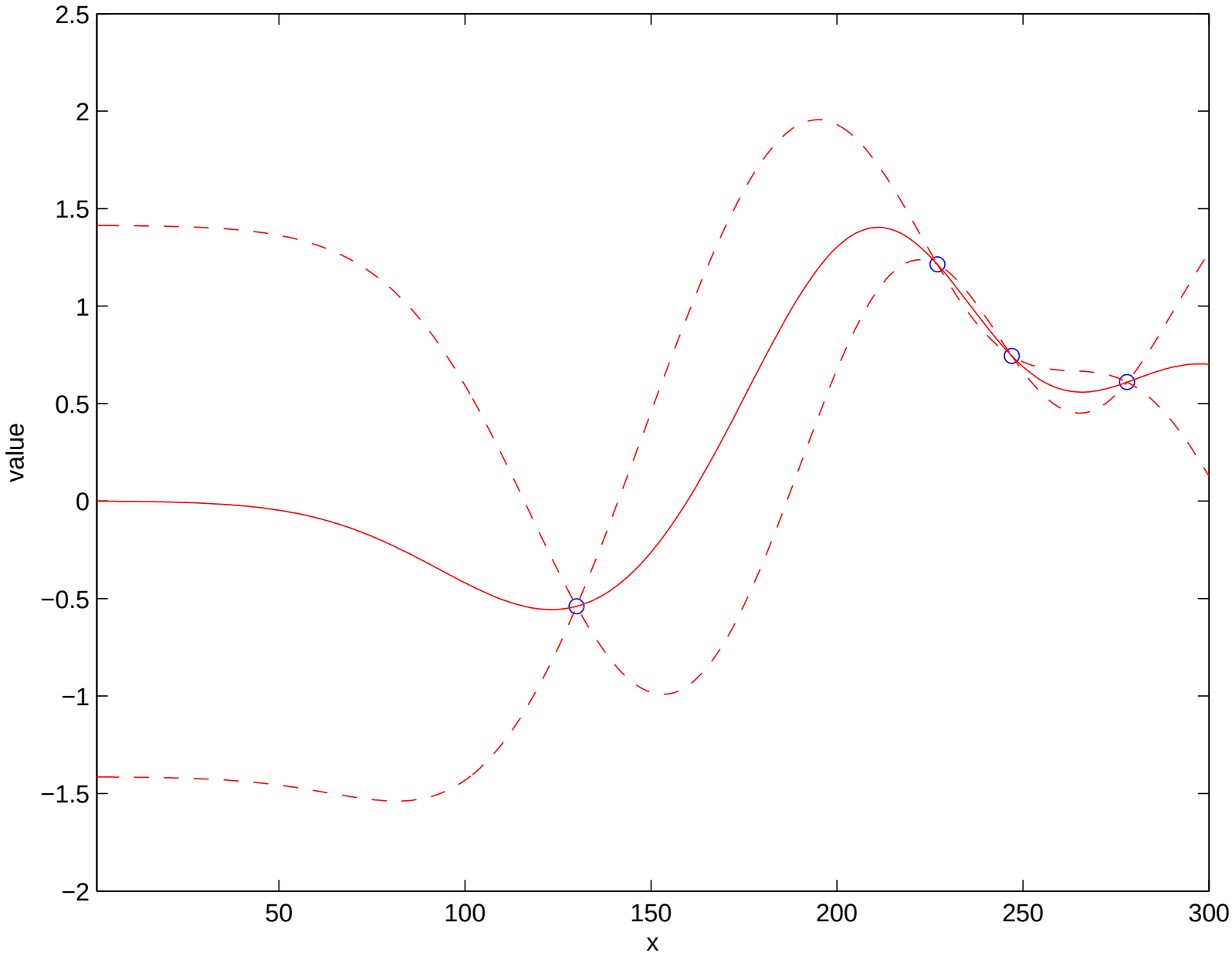
Illustrative 1D Example



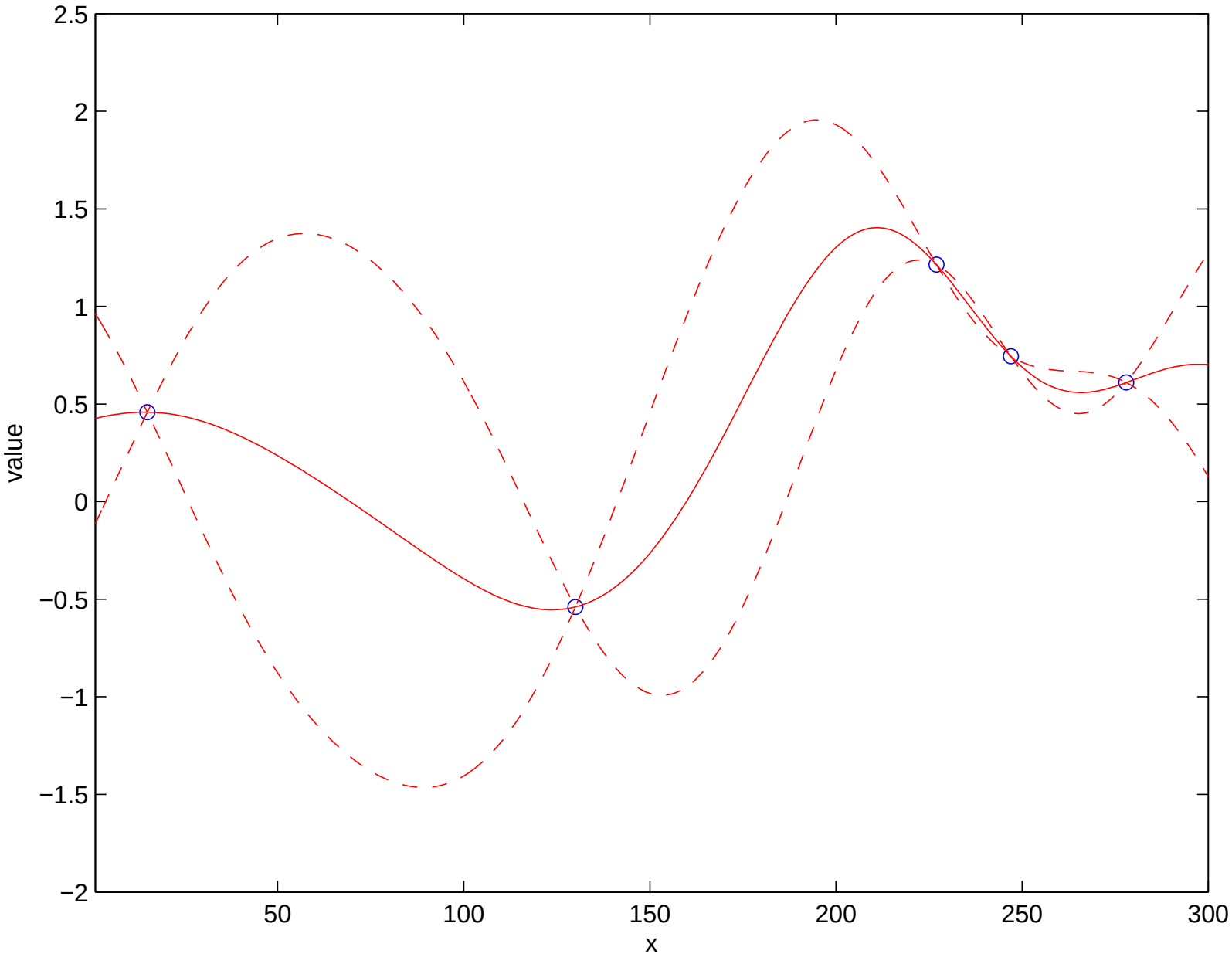
Illustrative 1D Example



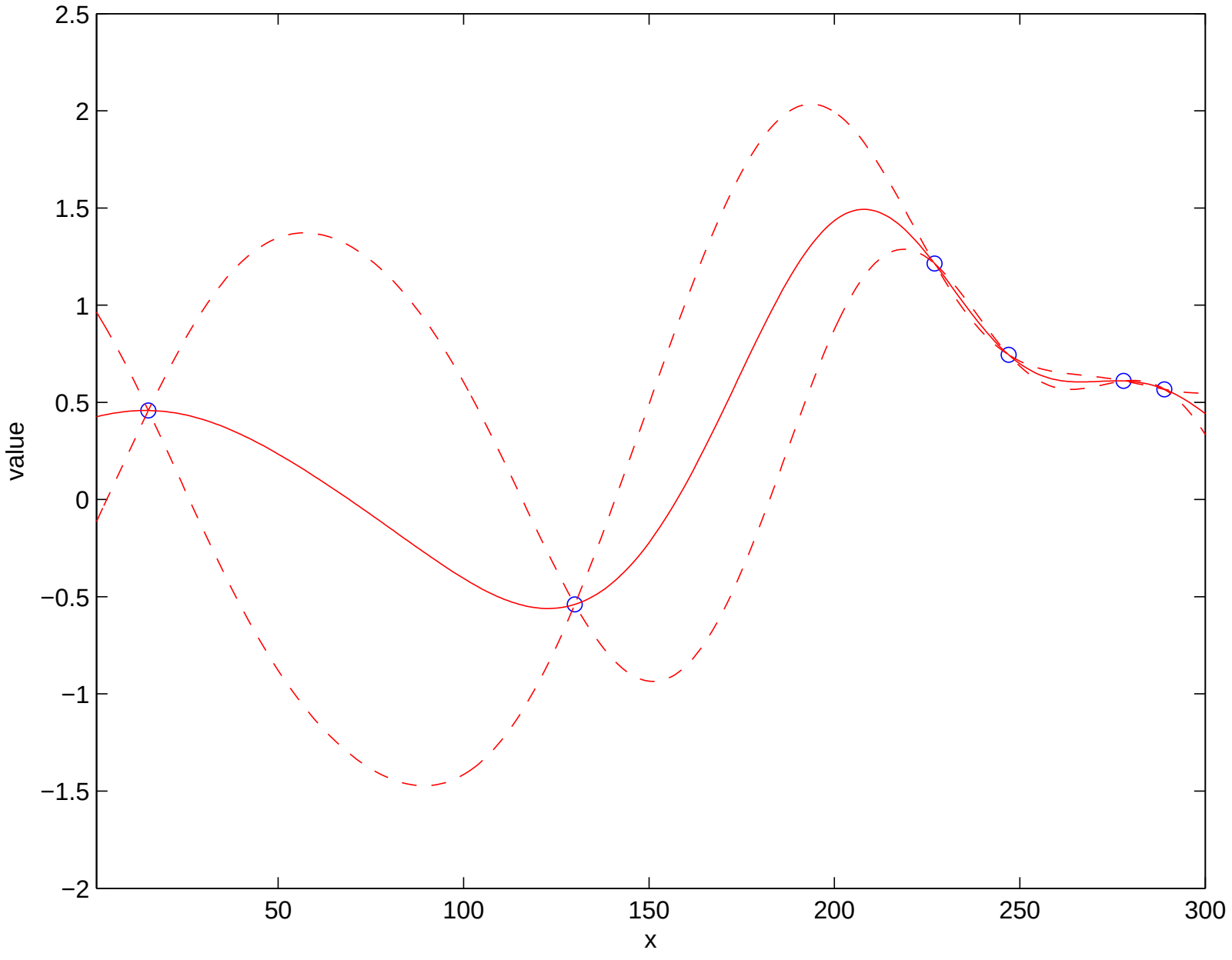
Illustrative 1D Example



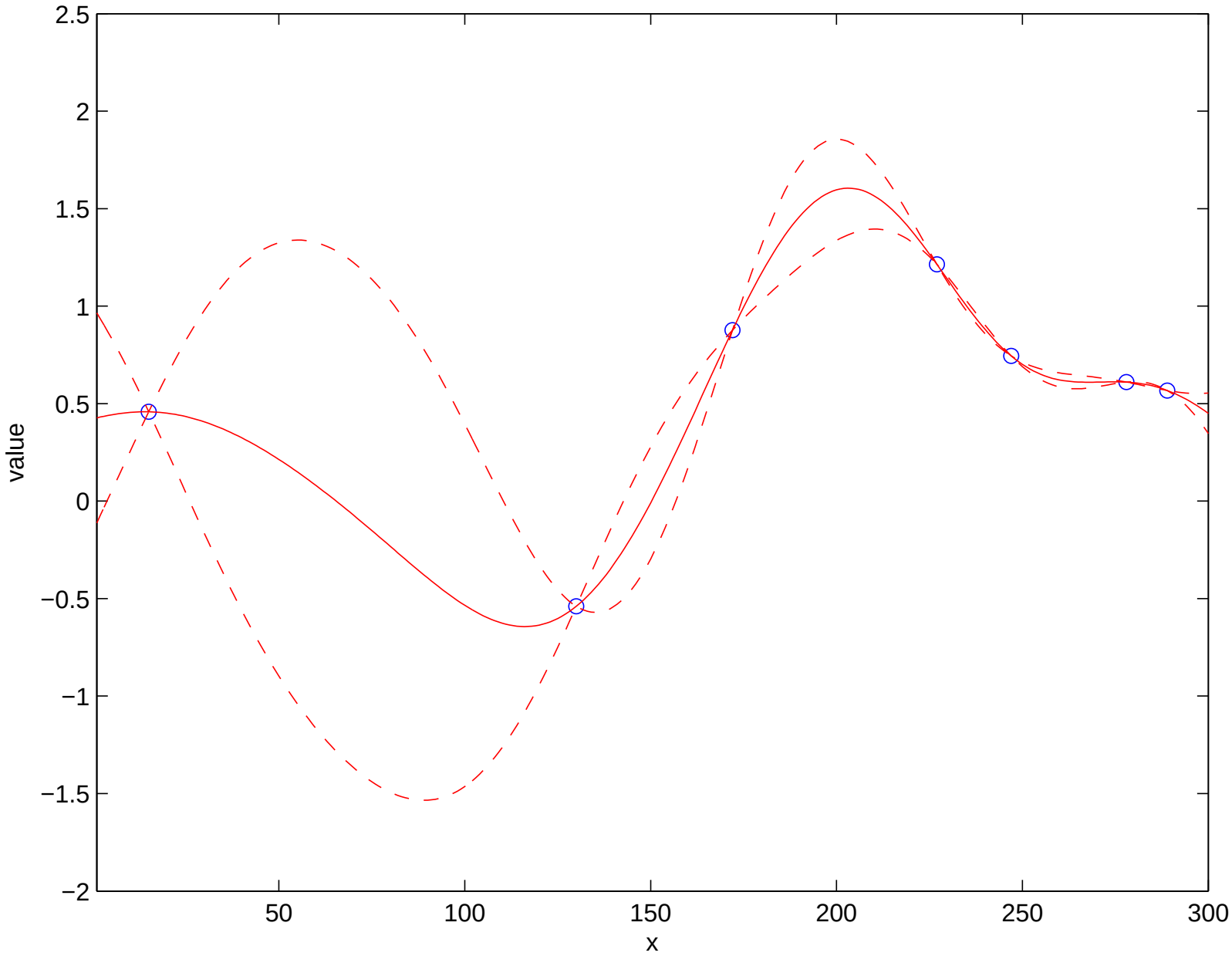
Illustrative 1D Example



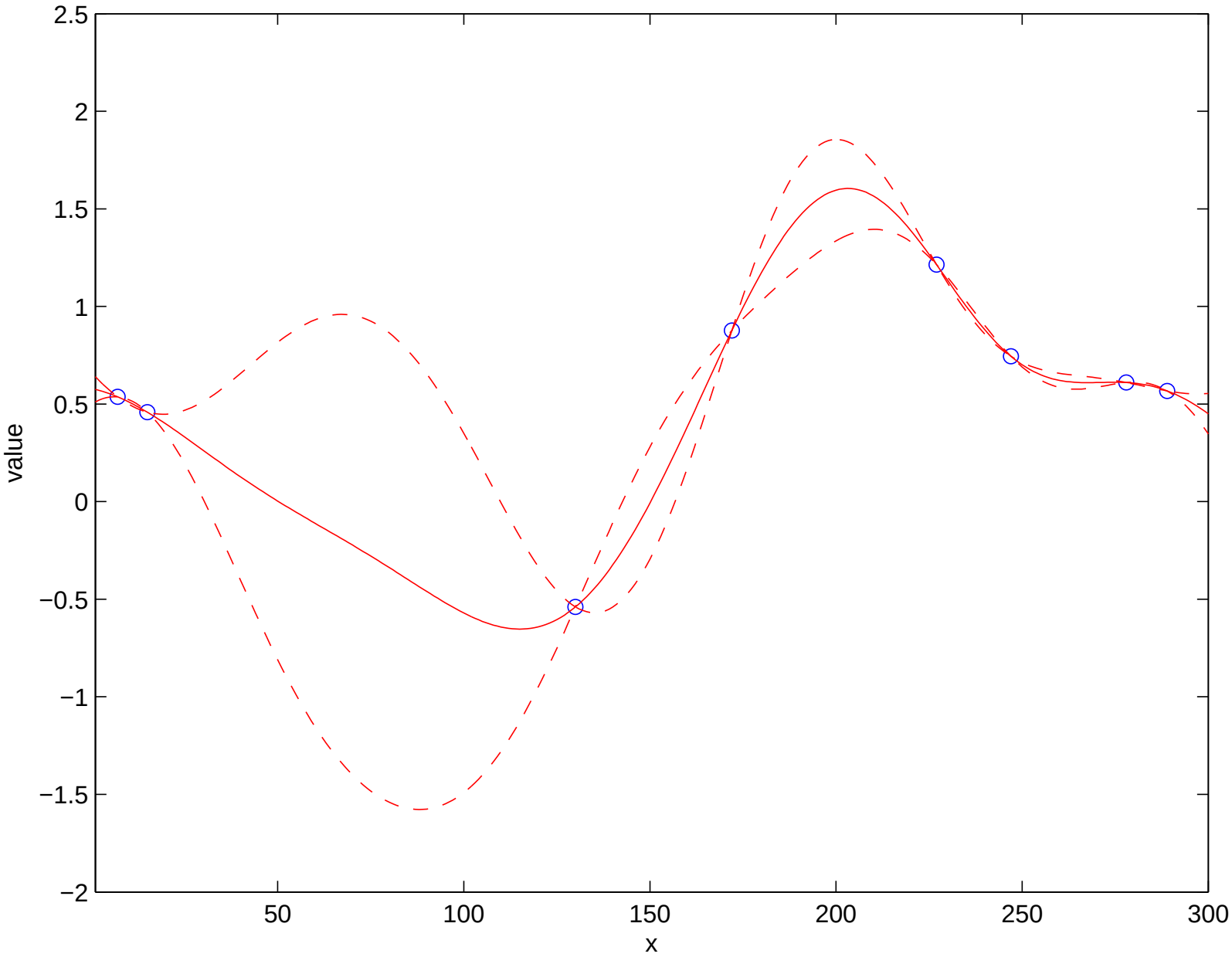
Illustrative 1D Example



Illustrative 1D Example



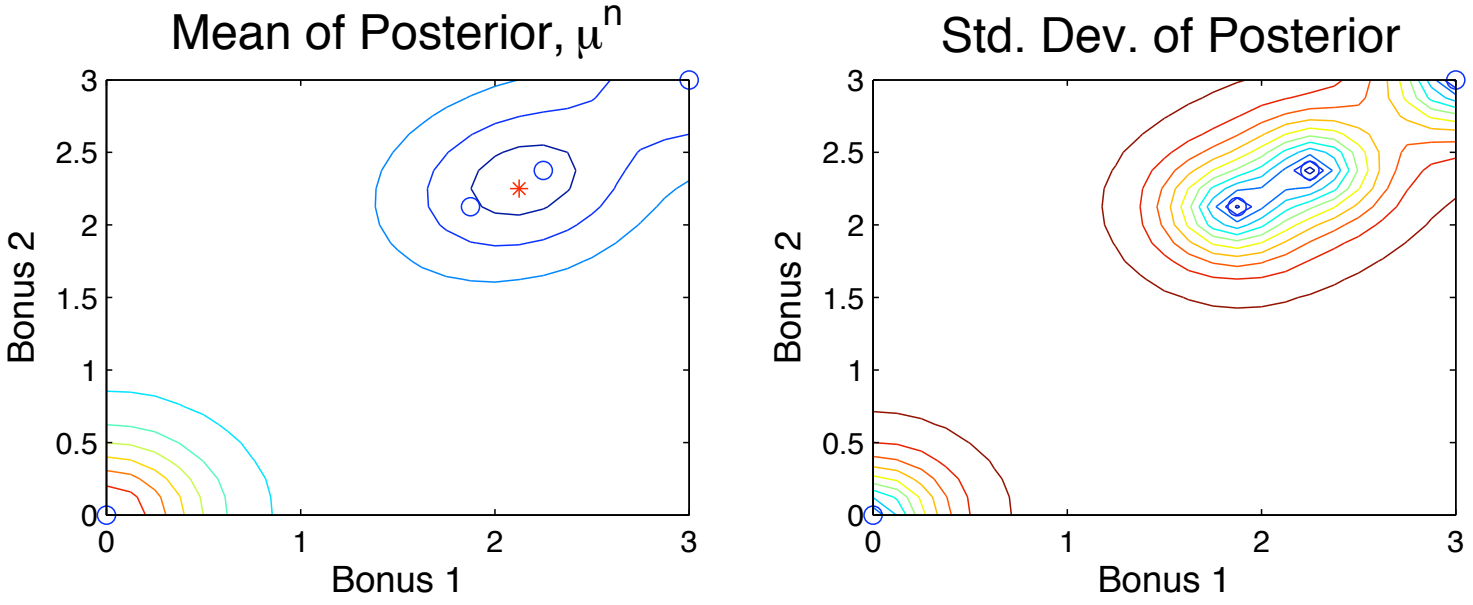
Illustrative 1D Example



GP Regression and BGO Work in More Than 1 Dimension

- For clarity, most of the illustrations in this talk are in 1-dimension.
- GP Regression and BGO can also be applied in $\mathbb{R}^d$ with $d > 1$, and also in combinatorial spaces.
- The key is including a notion of distance in $\Sigma_0(\cdot,\cdot)$, e.g.

$$\Sigma_0(x, x') = \alpha_0 \exp\left(-\alpha_1 ||x - x'||^2\right)$$



Outline

Outline

Expected Improvement

- In BGO, we use the posterior distribution to decide where to sample next.
- One classic method is called "Efficient Global Optimization" (EGO), and is based on the idea of **Expected Improvement**.
- This method is due to [Jones et al., 1998], building on ideas in [Mockus, 1972].

Expected Improvement

- Suppose we've measured n points $x_1, \ldots, x_n$, and observed $f(x_1), \ldots, f(x_n)$.
- Let $f_n^* = \max_{m=1,\ldots,n} f(x_m)$ be the best point observed so far.
- If we measure at a new point x, the **improvement** in our objective function is

$$[f(x) - f_n^*]^+$$

- The expected improvement is

$$\mathrm{EI}_n(x) = \mathbb{E}_n \left[[f(x) - f_n^*]^+ \right],$$

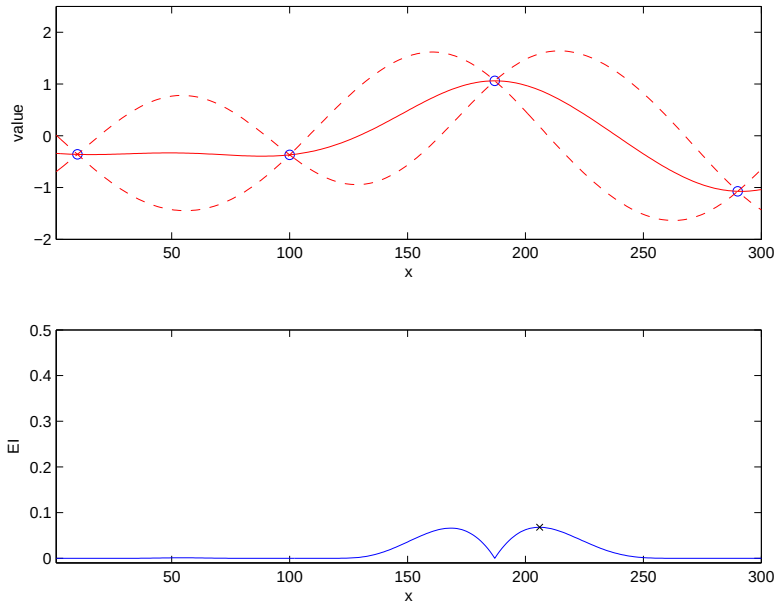
where $\mathbb{E}_n$ indicates the expectation taken with respect to the time-n posterior distribution.

Expected Improvement Can Be Computed Analytically

- Let $\Delta_n(x) = \mu_n(x) - f_n^*$ be the difference between our estimate of $f(x)$ and the best value observed so far. Then,

$$\begin{aligned}\mathrm{EI}_n(x) &= \mathbb{E}_n\left[[f(x) - f_n^*]^+\right] \\ &= [\Delta_n(x)]^+ + \sigma_n(x)\varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(-\frac{|\Delta_n(x)|}{\sigma_n(x)}\right),\end{aligned}$$

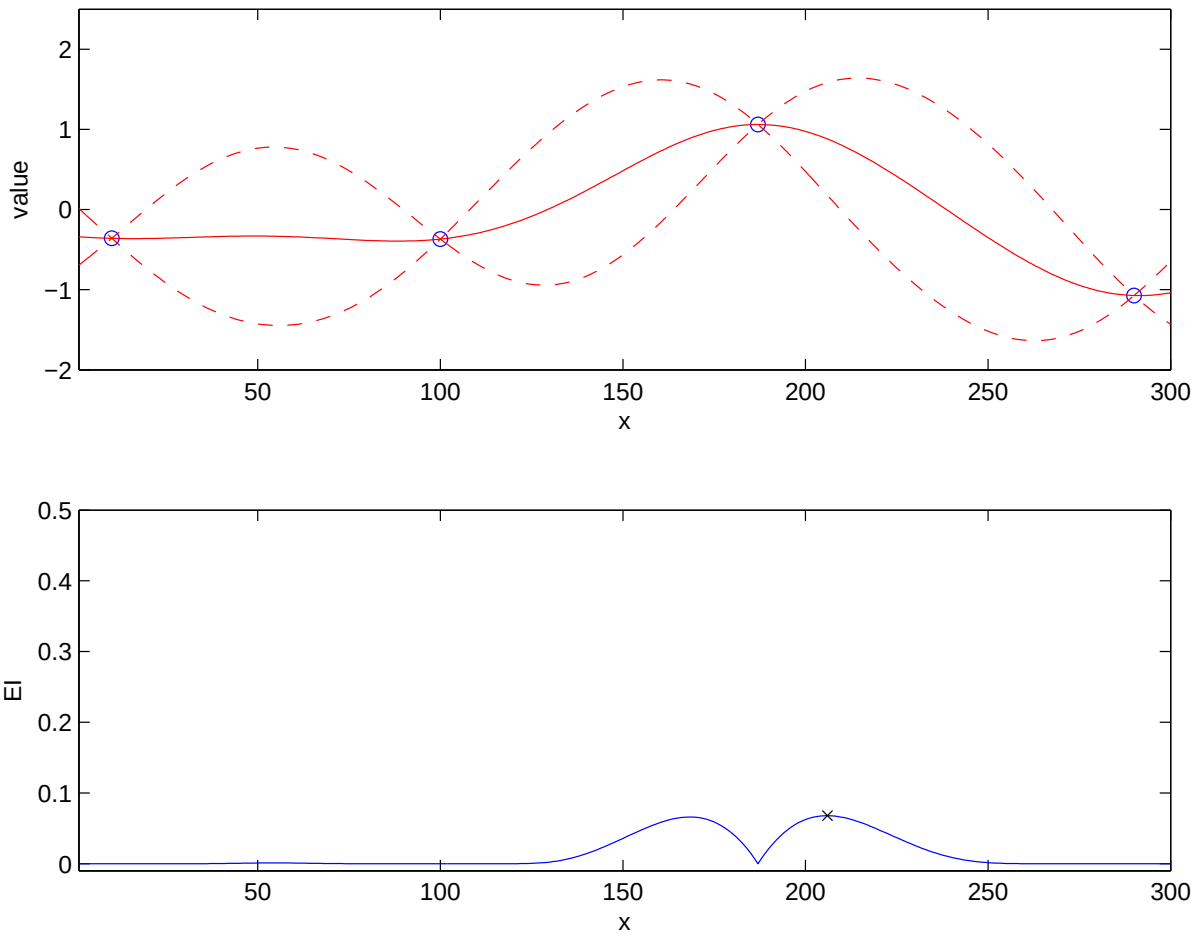
where Φ and φ are the normal cdf and pdf,



Expected Improvement

- The EGO/EI policy chooses to sample at the point with the largest expected improvement,

$$x_{n+1} = \arg\max_{x} \mathrm{EI}_n(x)$$



Expected Improvement

- The EGO/EI policy chooses to sample at the point with the largest expected improvement,

$$x_{n+1} = \arg\max_x \mathrm{EI}_n(x)$$

- Each time we decide which point to evaluate next (to solve our overall optimization problem), we have to solve an optimization problem!
- We have replaced one optimization problem ($\max_{x \in A} f(x)$) with many optimization problems ($\max_x \mathrm{EI}_n(x)$, for $n = 1, 2, 3, \ldots$). Why is this a good thing?
  - Evaluating $f(x)$ is expensive (minutes, hours, days), and derivative information is unavailable.
  - Evaluating $\mathrm{EI}_n(x)$ is quick (microseconds), and derivative information is available.

Maximize Expected Improvement

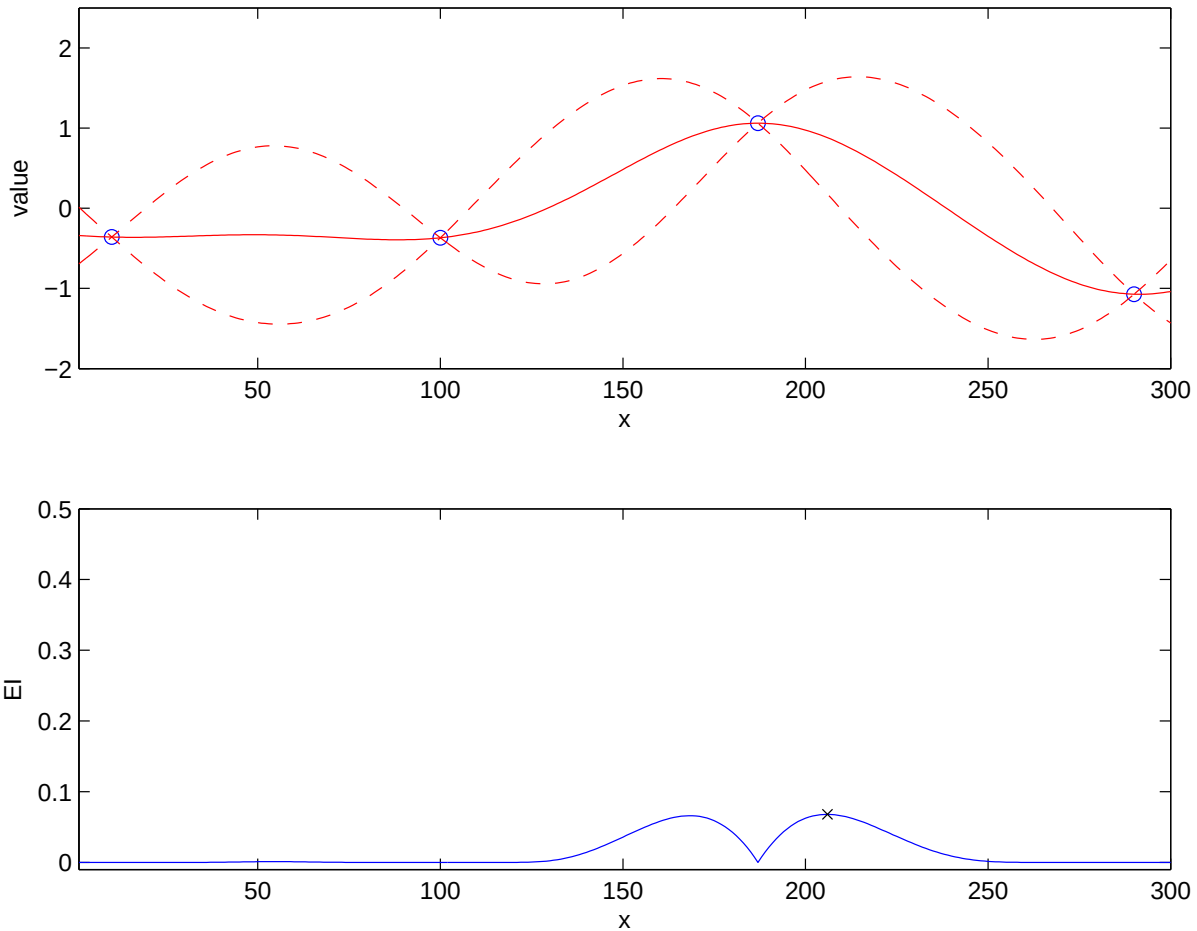
- The EGO/EI policy chooses to sample at the point with the largest expected improvement,

$$x_{n+1} = \arg\max_x \mathrm{EI}_n(x)$$

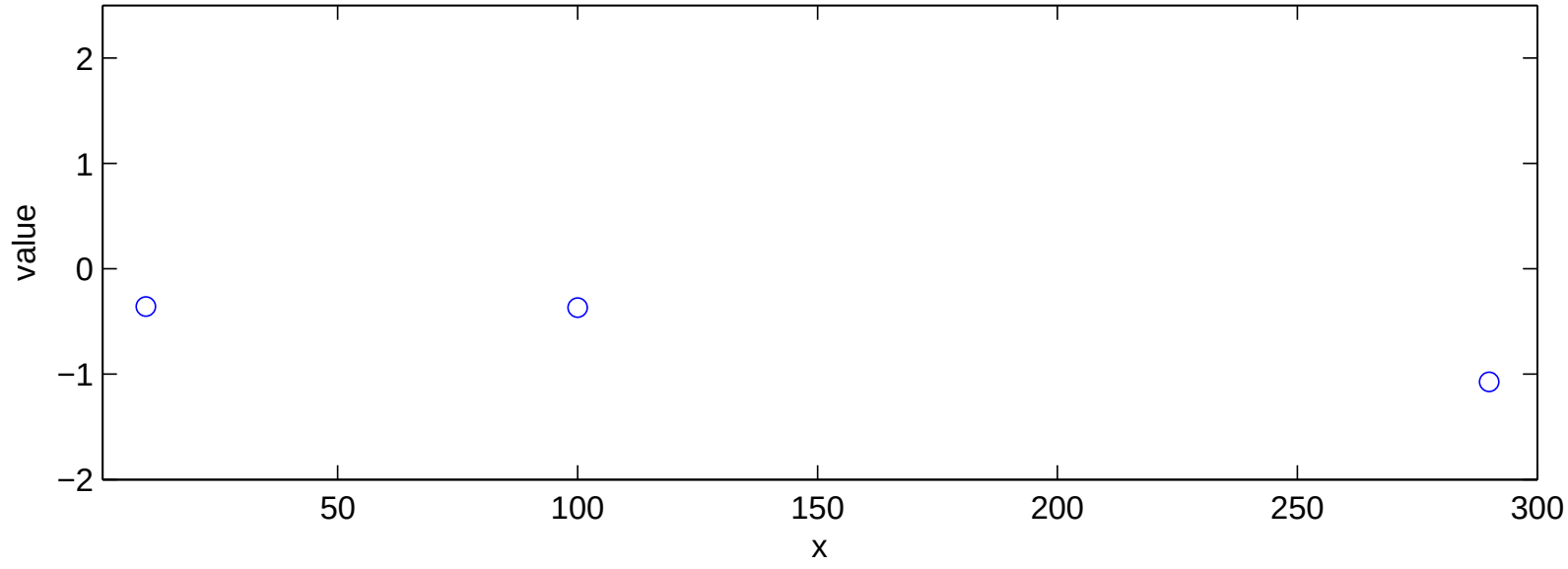
- One can calculate the gradient of $\mathrm{EI}_n(x)$ with respect to x.
- To solve $\max_x \mathrm{EI}_n(x)$, use a first order method combined with multistart.

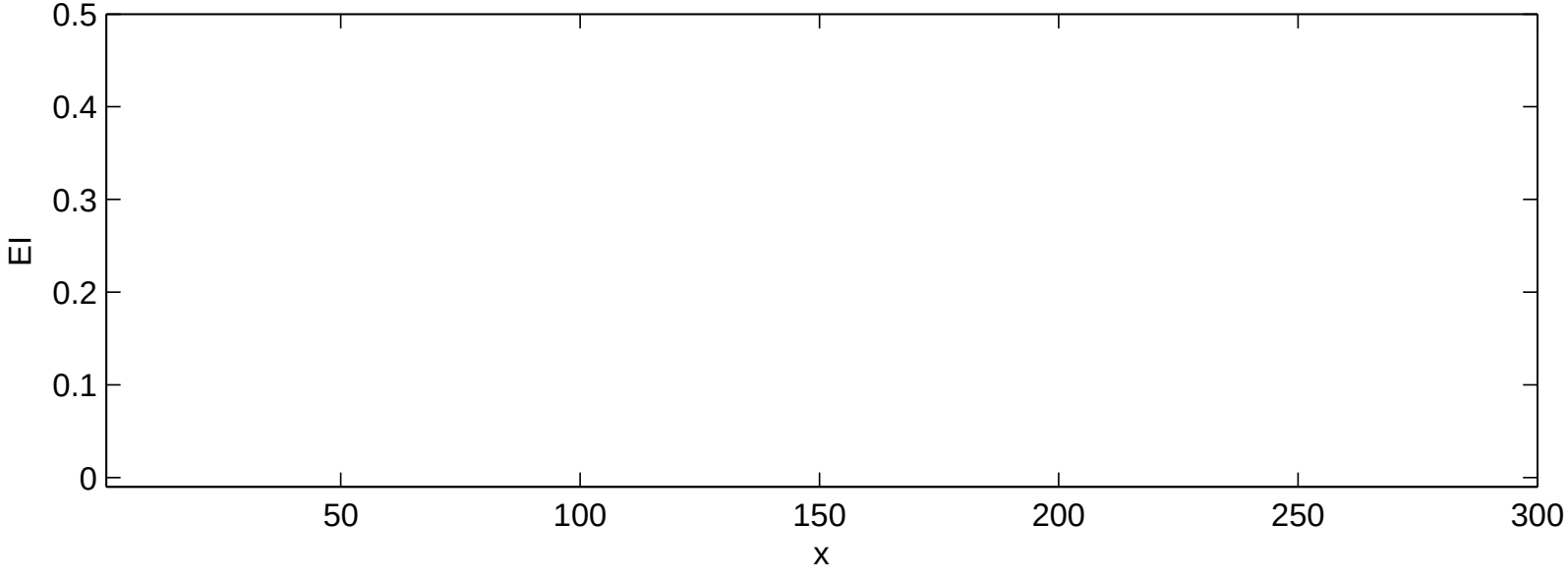
EI Trades Exploration vs. Exploitation

- $\mathrm{EI}_n(x)$ is bigger when $\mu_n(x)$ is bigger.
- $\mathrm{EI}_n(x)$ is bigger when $\sigma_n(x)$ is bigger.
- These two tendencies often push against each other, and the EI policy must balance them.

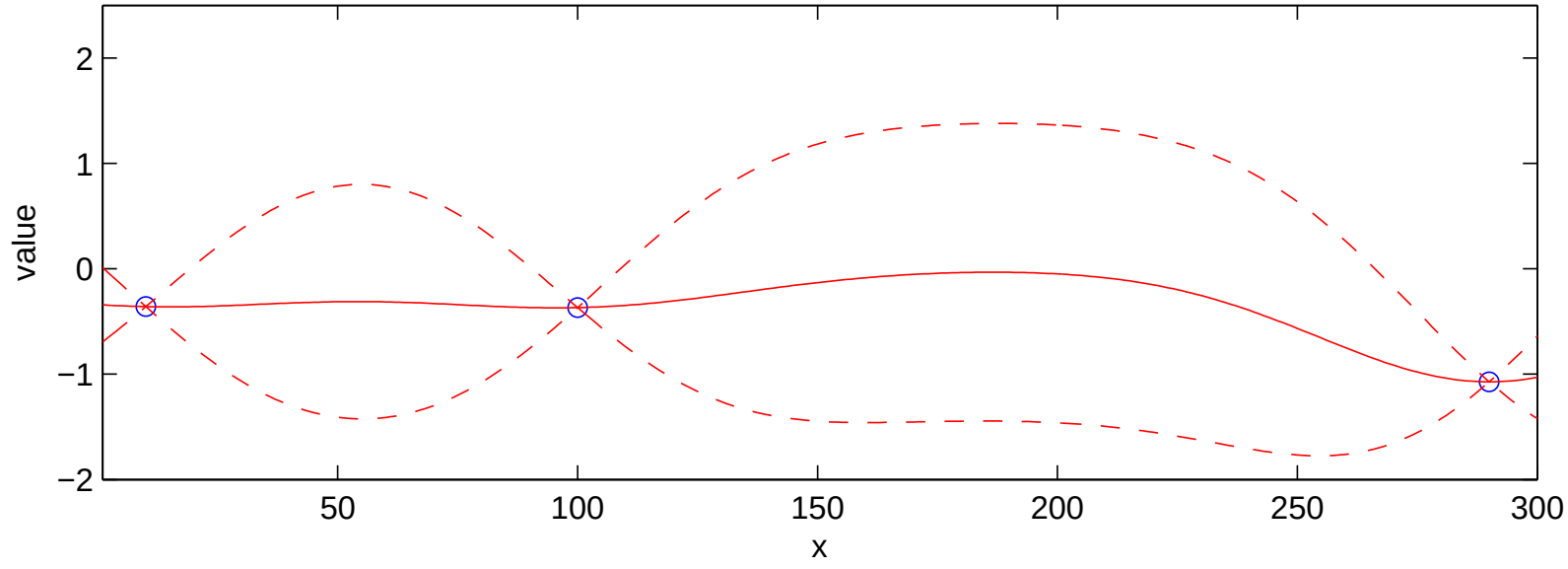


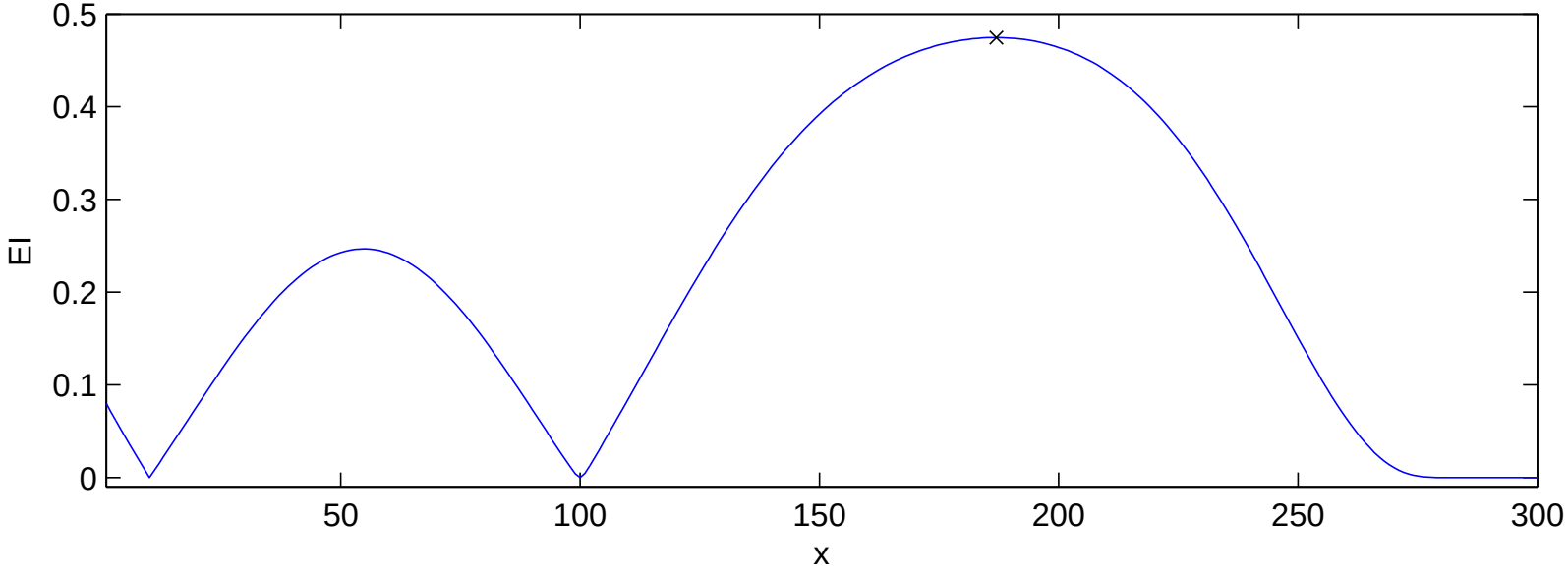
EGO Animation



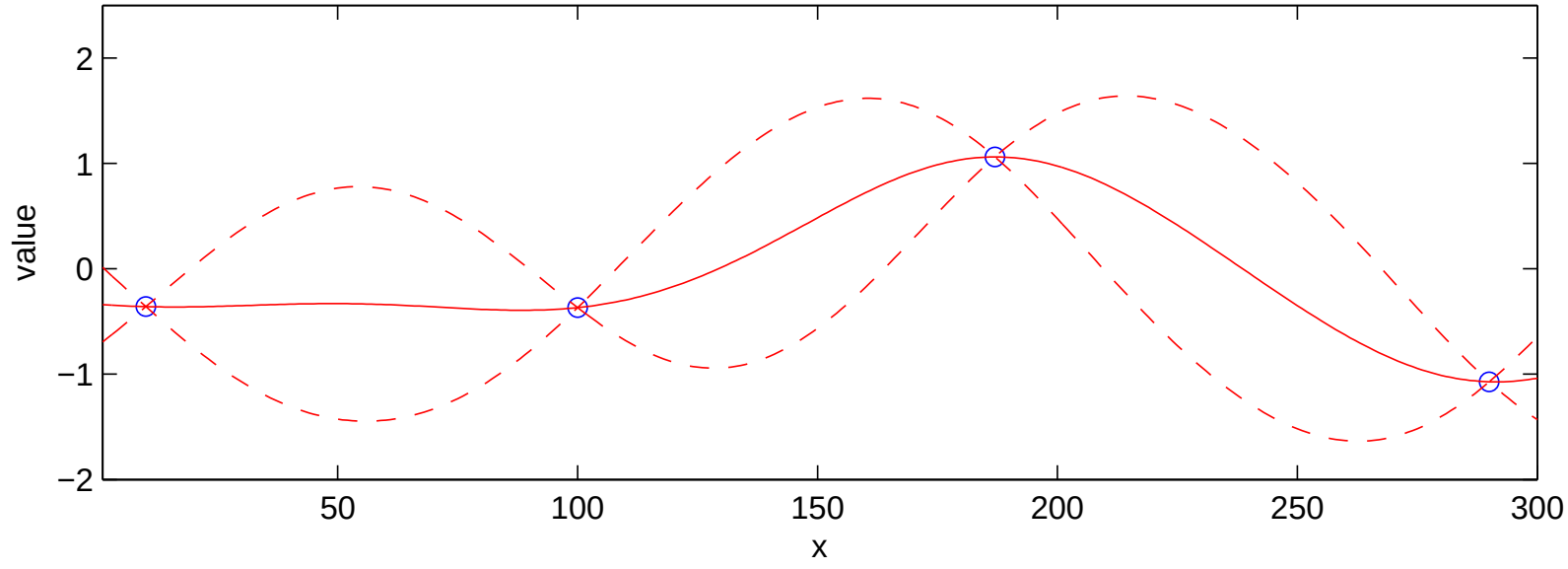


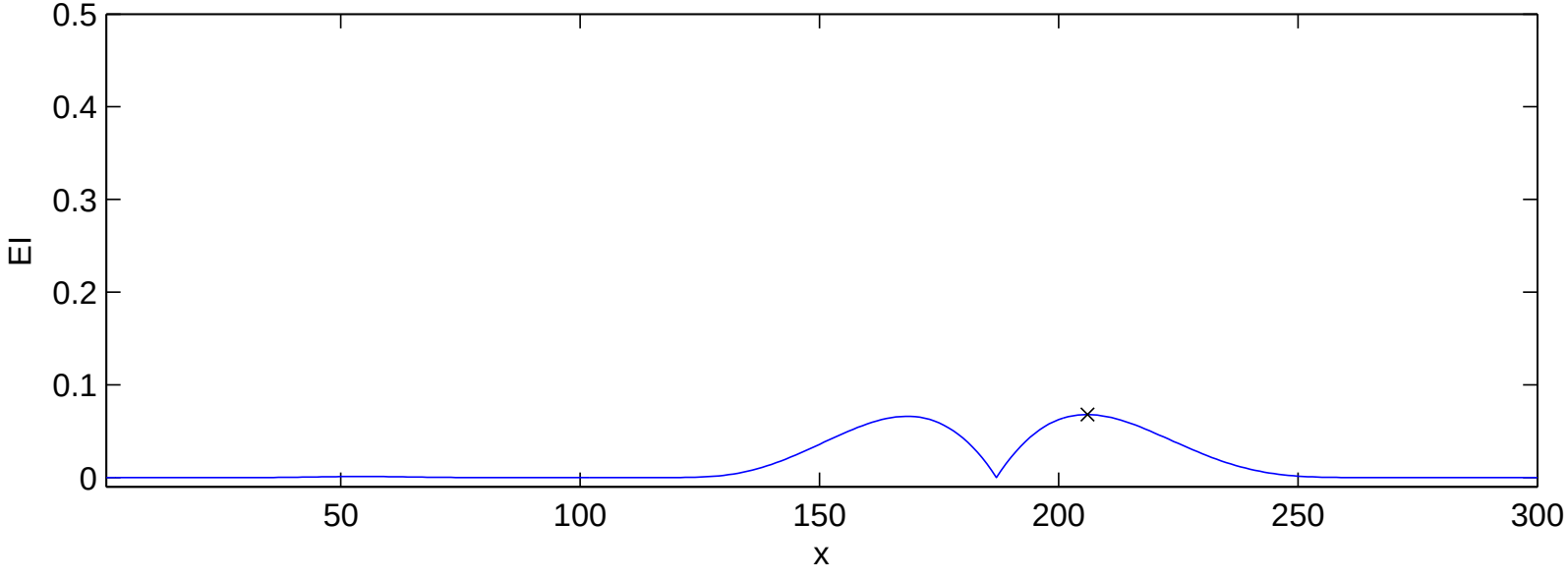
EGO Animation



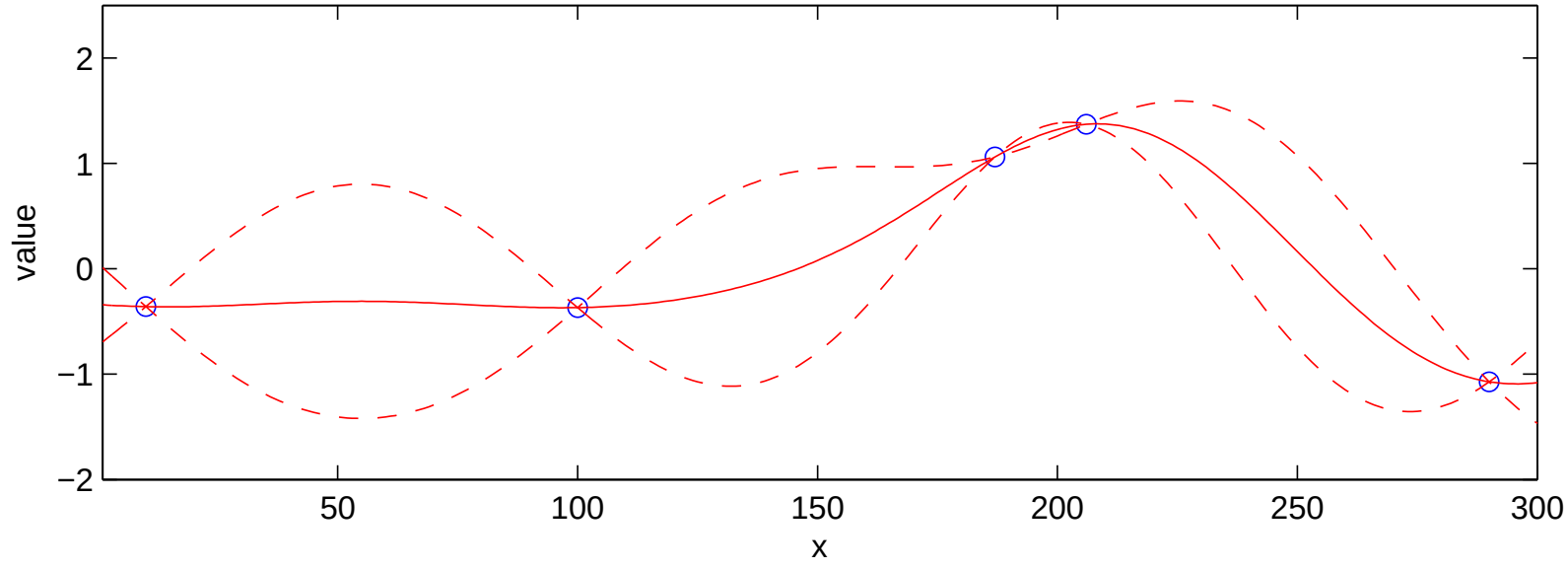


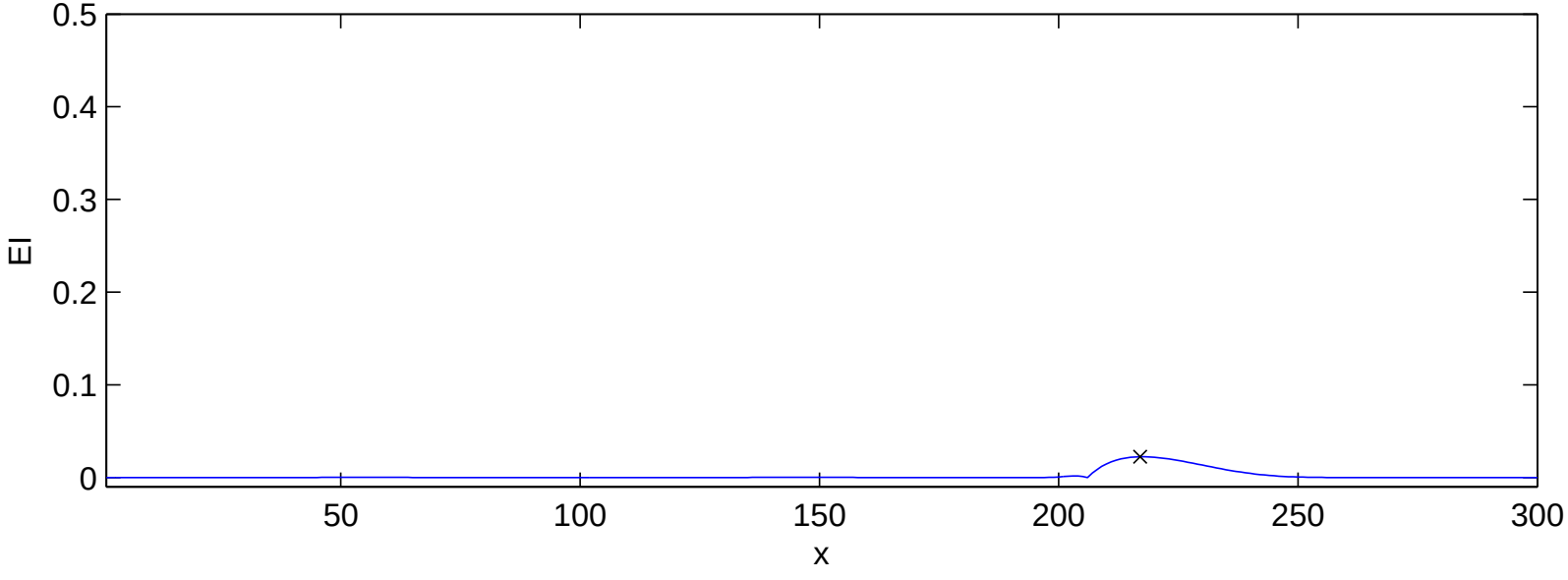
EGO Animation



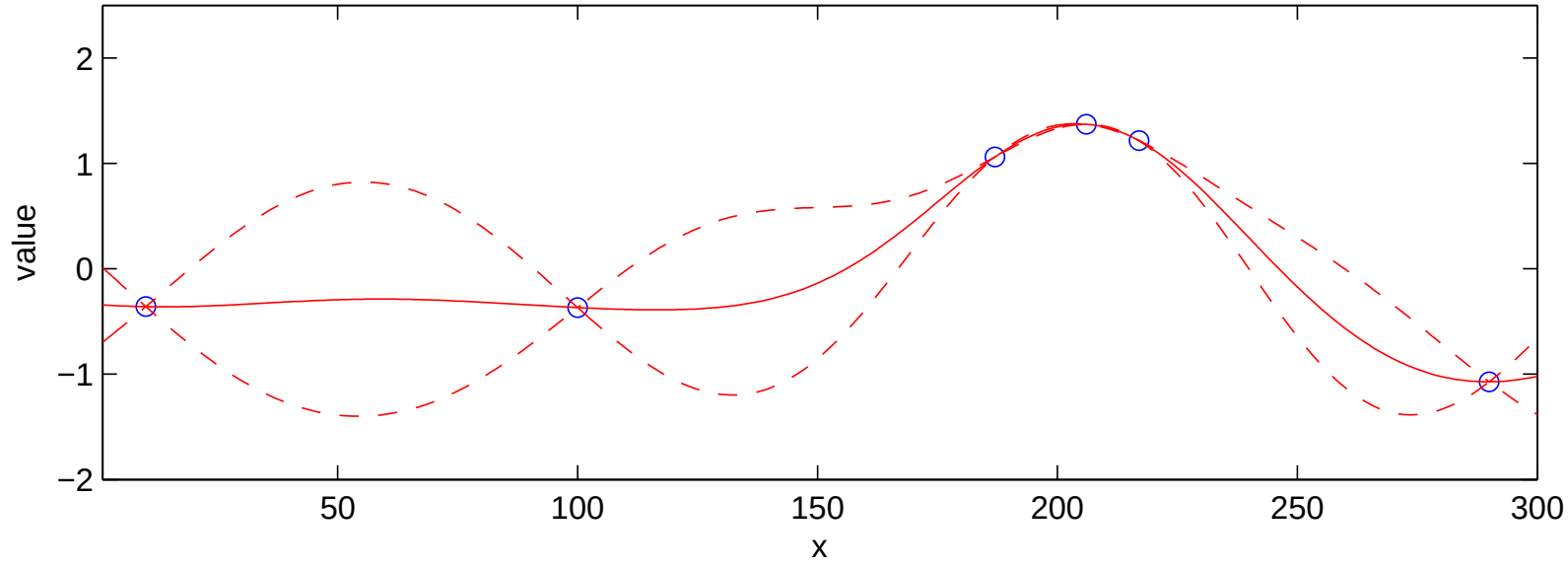


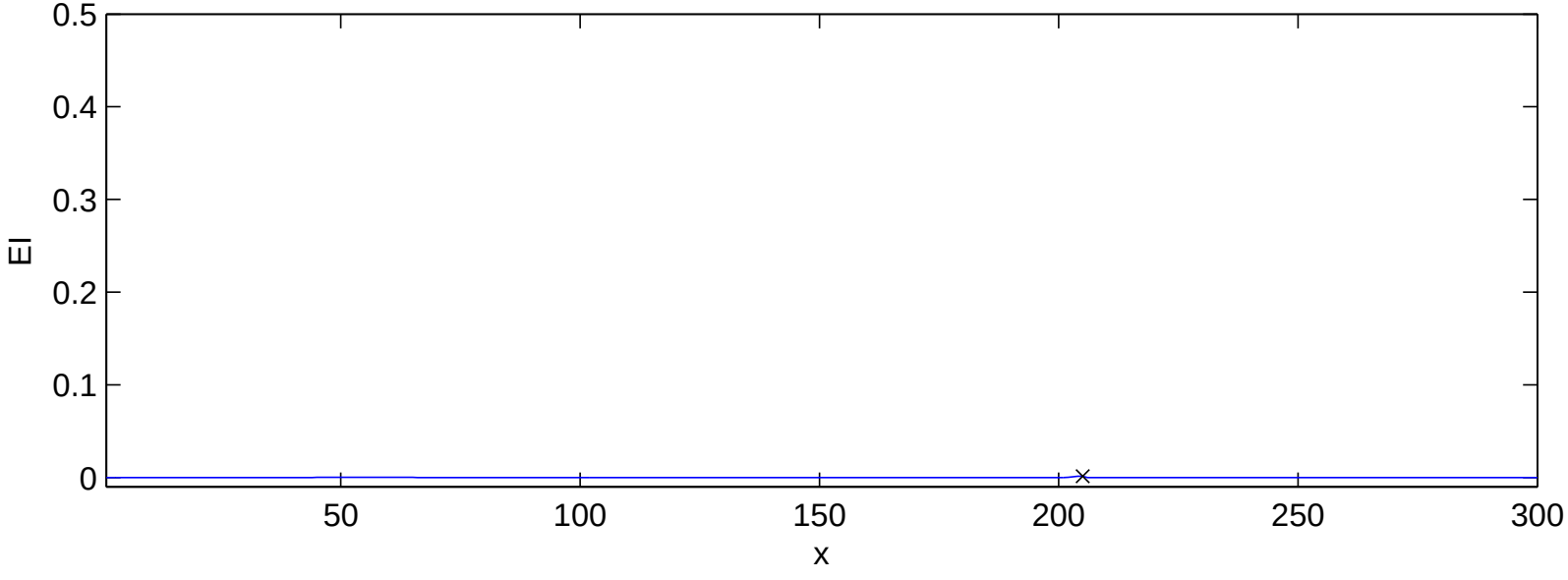
EGO Animation





EGO Animation





Outline

Requirement for Use: Expensive Function Evaluation

BGO is only useful when function evaluation is time-consuming or expensive.

- In the simulation calibration problem discussed later, each function evaluation takes 3 days.
- In the drug development problem discussed later, each function evaluation takes several days.
- How expensive is expensive enough? Function evaluation should take significantly longer than the time that the BGO algorithm requires to decide where to sample next.
- BGO takes longer to decide where to take each sample, but requires fewer samples than other methodologies (when it works well).

Requirement for Use: Lack of Gradient Information

- If gradient information is available, it is usually better to simply use a multistart first-order method.
- Gradient information can be incorporated into a BGO algorithm to improve its speed, but this is difficult and is not covered here.
- Incorporating gradient information into BGO algorithms remains an area for research.

Other Derivative-Free Global Optimization Methods

- Many other derivative-free noise-tolerant global optimization methods exist, e.g.,
  - pattern search, e.g., Nelder-Mead
  - stochastic approximation, e.g., SPSA [Spall 1992].
  - evolutionary algorithms, simulated annealing, tabu search
  - response surface methods. [Myers & Montgomery 2002]
  - Lipschitzian optimization, e.g., DIRECT [Gablonsky et al. 2001]
- BGO methods require more computation to decide where to evaluate next, but require fewer evaluations to find global extrema (caveat: when the prior is chosen well).
  - [Huang et al. 2006] compares sequential kriging optimization (a BGO method) against DIRECT 0Gablonsky et al 2001], Nelder-Mead modified for noise by Humphrey et al 2000, and SPSA [Spall 1992], and finds that SKO requires fewer function evaluations.

BGO is a Surrogate Method

- BGO methods operate by maintaining a posterior distribution on the unknown objective function f,
- There is a class of global optimization methods called **surrogate methods** that maintain a cheap-to-evaluate approximation to the objective function, and use this to decide where to sample next. (see, e.g., [Booker et al., 1999, Regis and Shoemaker, 2005])
- The mean of the posterior distribution can be thought of as a surrogate, and so, loosely speaking, BGO methods are a type of surrogate method.

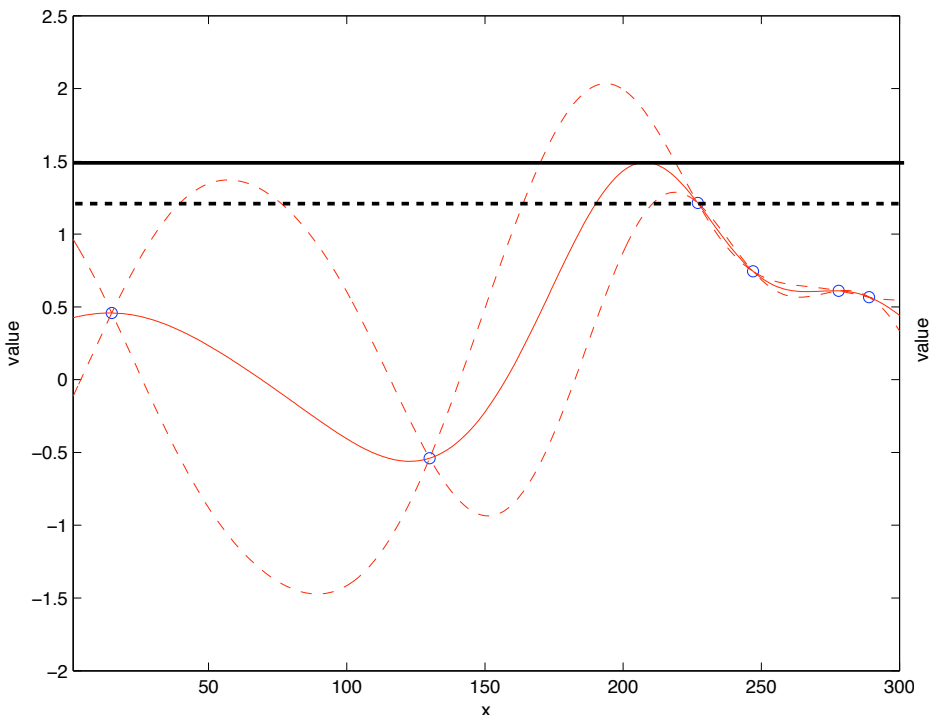
Outline

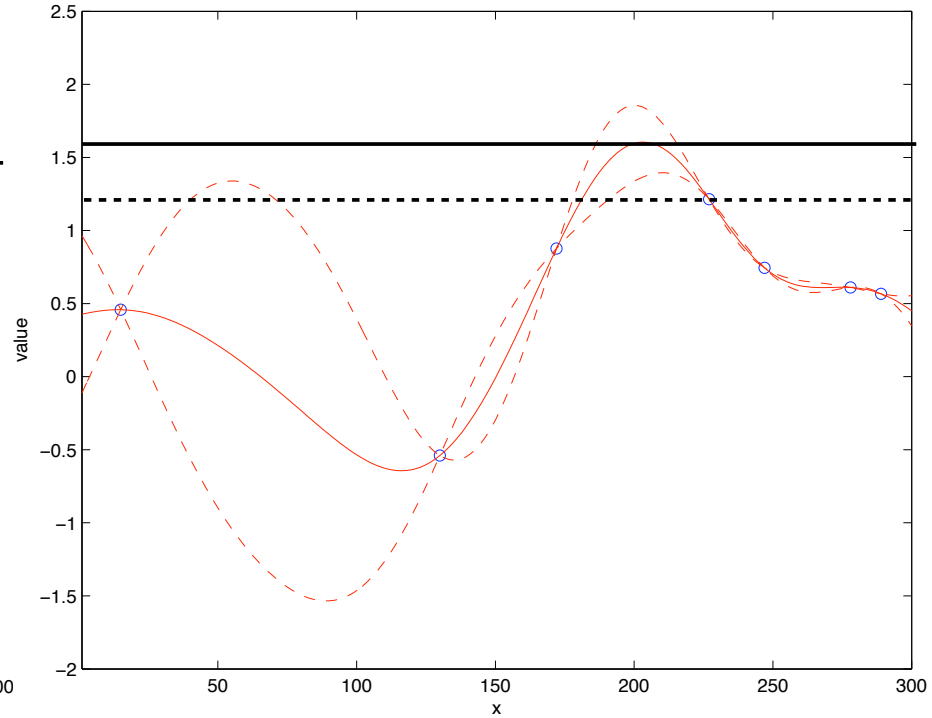
Best estimated overall value might be at an unmeasured point

- The improvement considered by EI is:

$$[f(x) - f_n^*]^+ = \max(f(x), f_n^*) - f_n^* = f_{n+1}^* - f_n^*$$

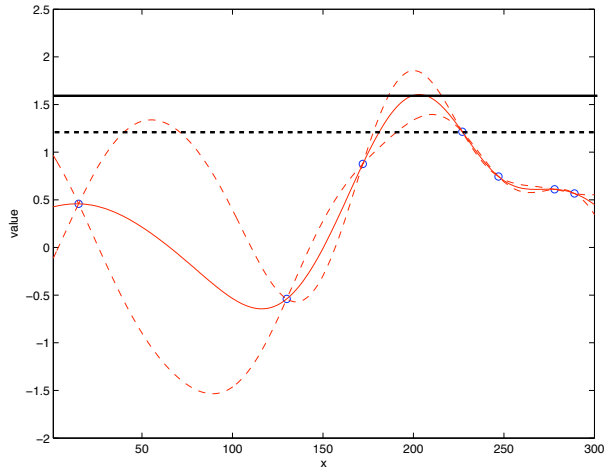
  where $f_n^* = \max_{m \le n} f(x_m)$ is the best point we've measured by time n.
- But the point with the best estimated value might not be a point we've measured.





We can measure improvement w.r.t. the best overall value

Replace $f_n^* = \max_{m \le n} f(x_m) = \max_{m \le n} \mu_n(x_m)$ with $\mu_n^* = \max_{x \in A} \mu_n(x)$.



- The corresponding improvement is $\mu_{n+1}^* - \mu_n^*$.
- The corresponding value for taking a sample is

$$\mathbb{E}_n \left[\mu_{n+1}^* - \mu_n^* \mid x_{n+1} = x \right].$$

- The policy that measures at the x with the largest such value is called the **knowledge-gradient with correlated beliefs** (KGCB) policy.

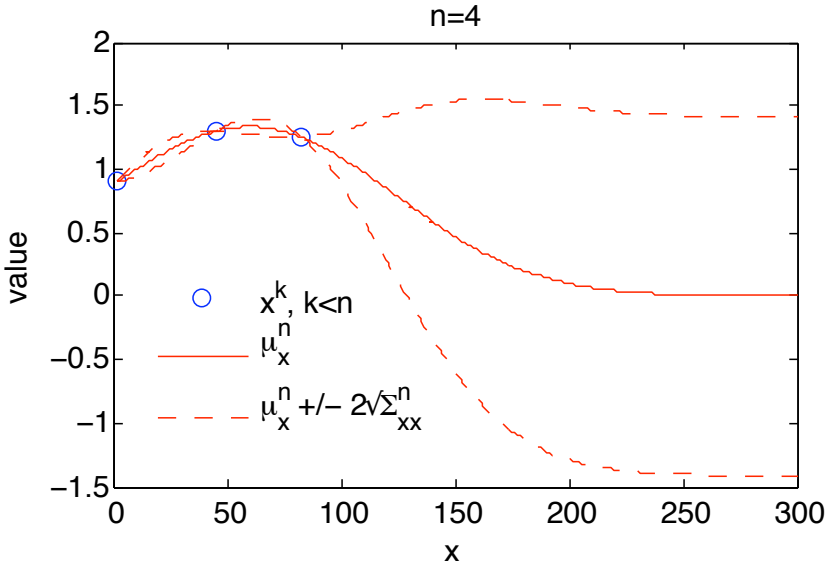
Knowledge-Gradient with Correlated Beliefs (KGCB)

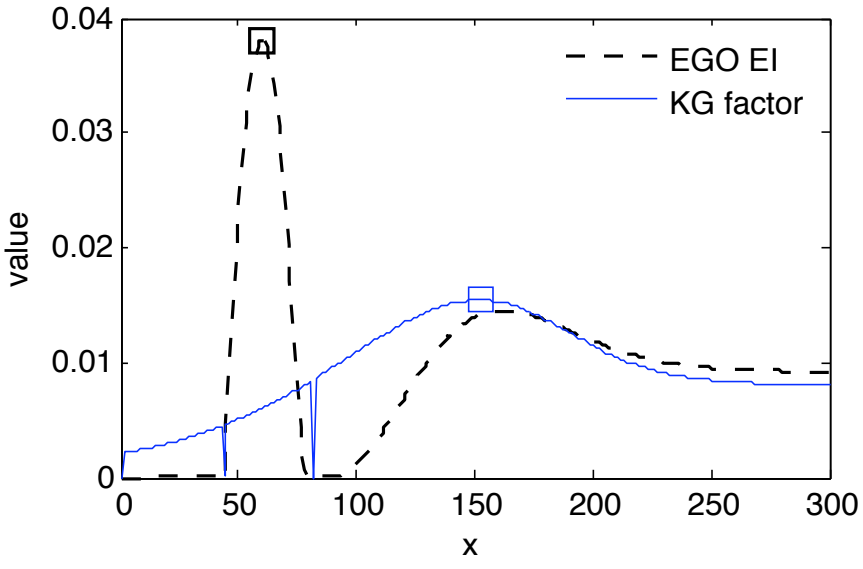
- Call this modified expected improvement the knowledge-gradient (KG) factor

$$\mathrm{KG}_n(x) = \mathbb{E}_n\left[\mu^*_{n+1} - \mu^*_n \mid x_{n+1} = x\right].$$

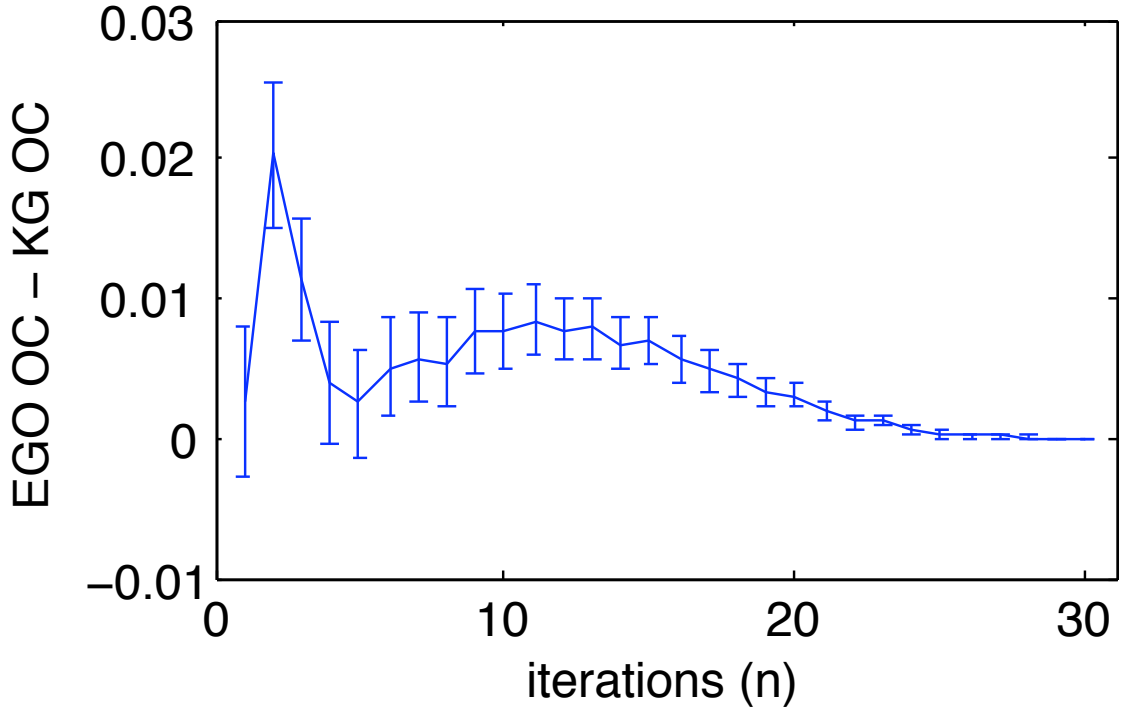
- The KGCB policy measures at the point with the largest KG factor.

$$x_{n+1} \in \arg\max_x \mathrm{KG}_n(x).$$





KGCB Requires Fewer Function Evaluations than EGO, but More Computation



- Graph shows the difference in expected solution quality between KGCB and EGO, on noise-free problems.
- KGCB needs fewer function evaluations to find a good solution, but more computation to decide where to evaluate.

Outline

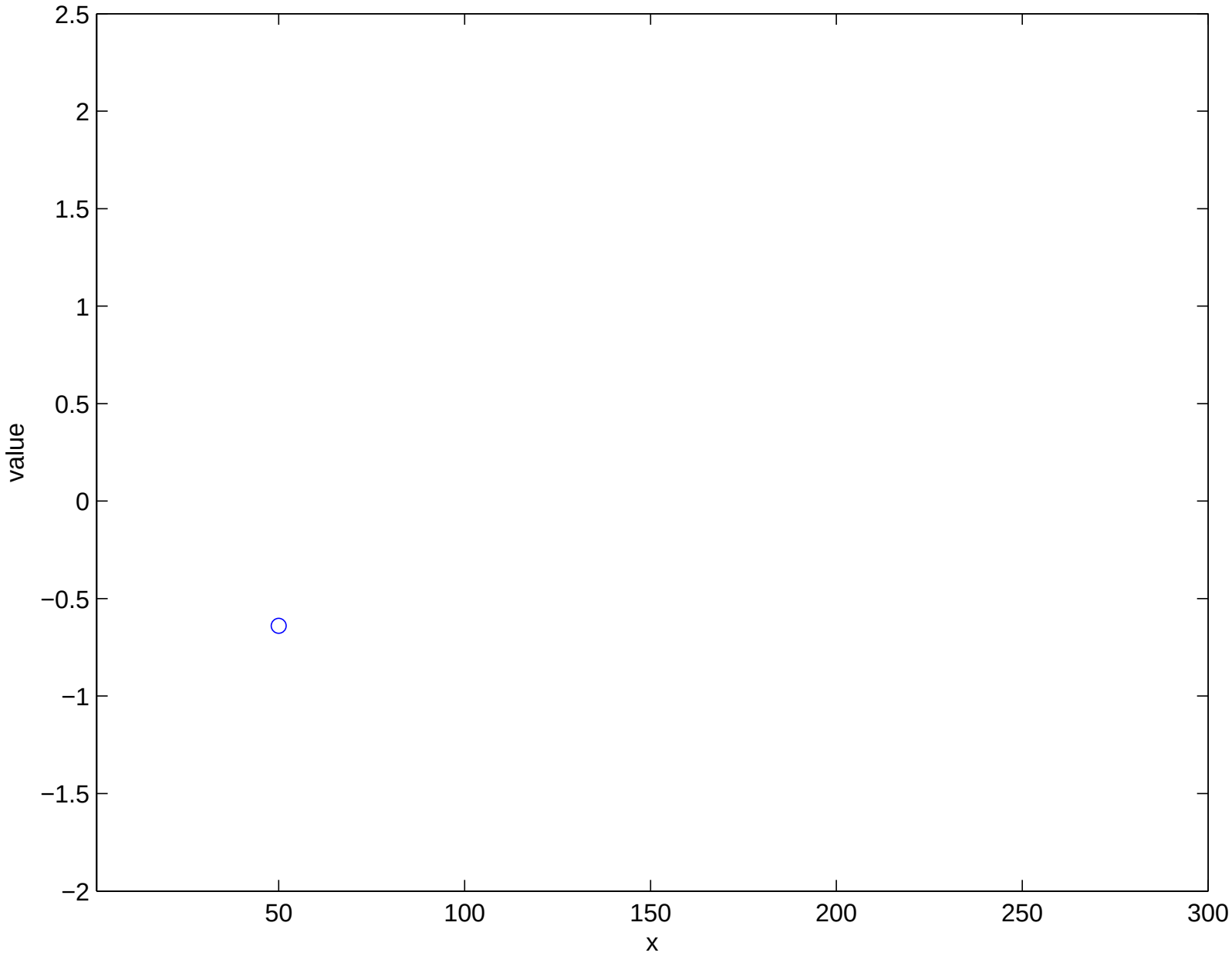
Noisy Global Optimization

- Thus far we have assumed noise-free function evaluations $f(x)$.
- What if we observe function evaluations with noise, $g(x, \omega)$?
- We use the same approach:
  1. Use GP regression to calculate the posterior on $f(x) = \mathbb{E}[g(x, \omega)]$ from noisy function evaluations.
  2. Use the posterior to decide where to sample next.
  3. Repeat.

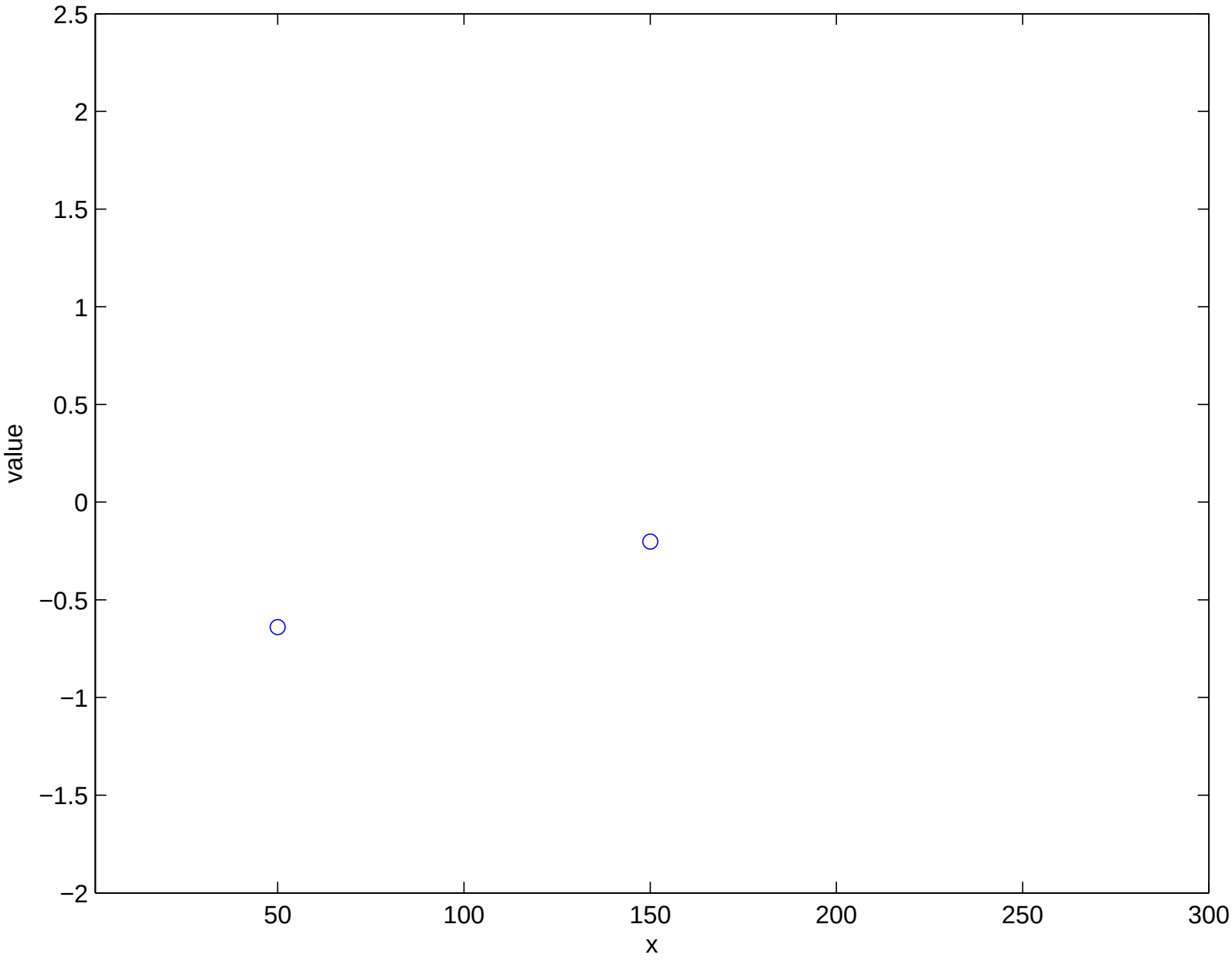
GP Regression Can Be Generalized to Allow Noise

- What if we have noisy measurements? i.e., we observe $g(x, \omega) = f(x, \omega) + \varepsilon(x, \omega)$.
- If the noise is normally distributed with a known (possibly heterogeneous) variance, then we can still calculate the posterior in essentially the same way.
- In practice, the noise is neither normal nor of known variance, but it remains a useful approximation. (In practice, one estimates the variance as you go.)
- Current research examines what can be done to get rid of this approximation. (e.g., stochastic kriging from [Ankenman, Nelson and Staum 2010])

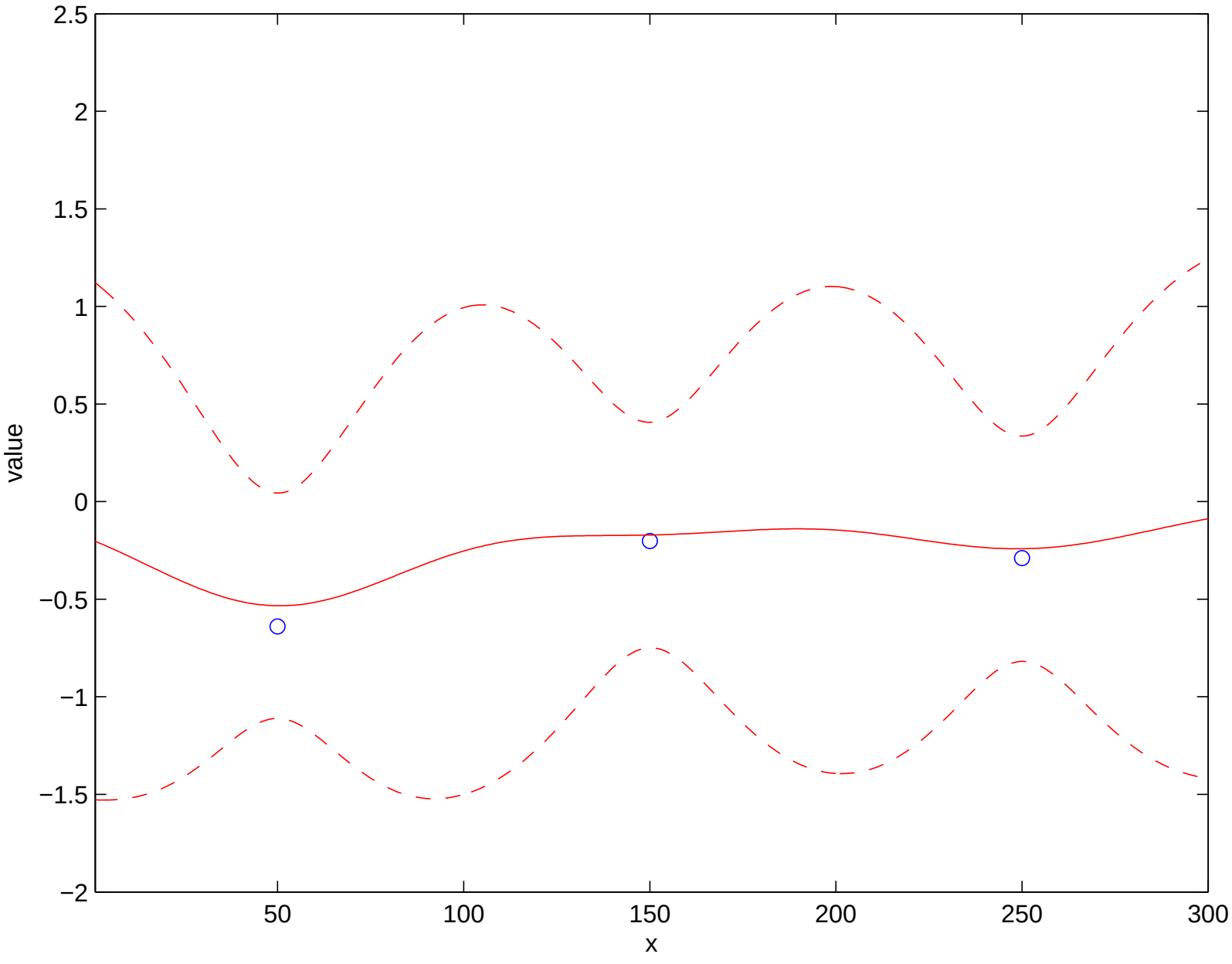
Illustrative 1D Example with Noise



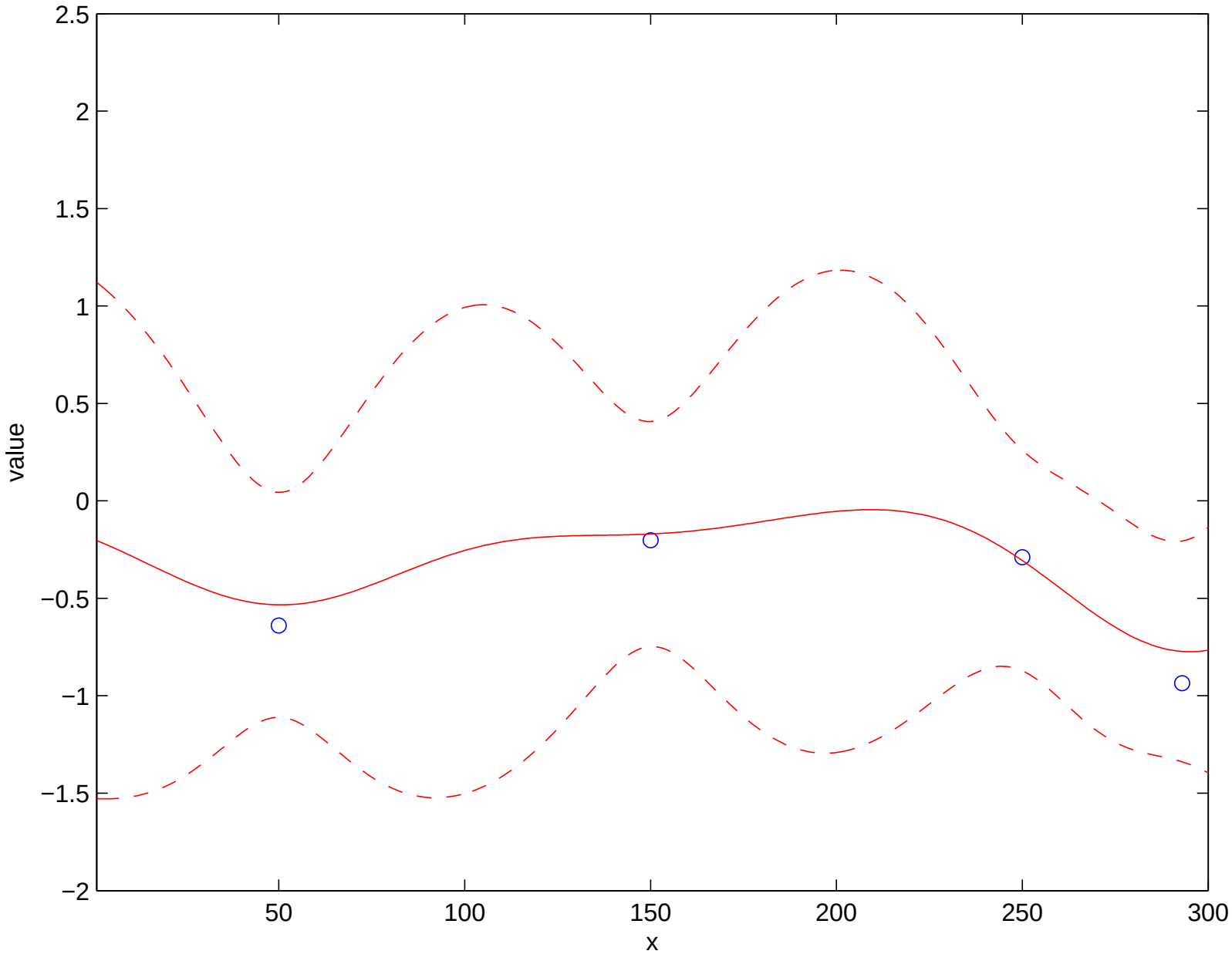
Illustrative 1D Example with Noise



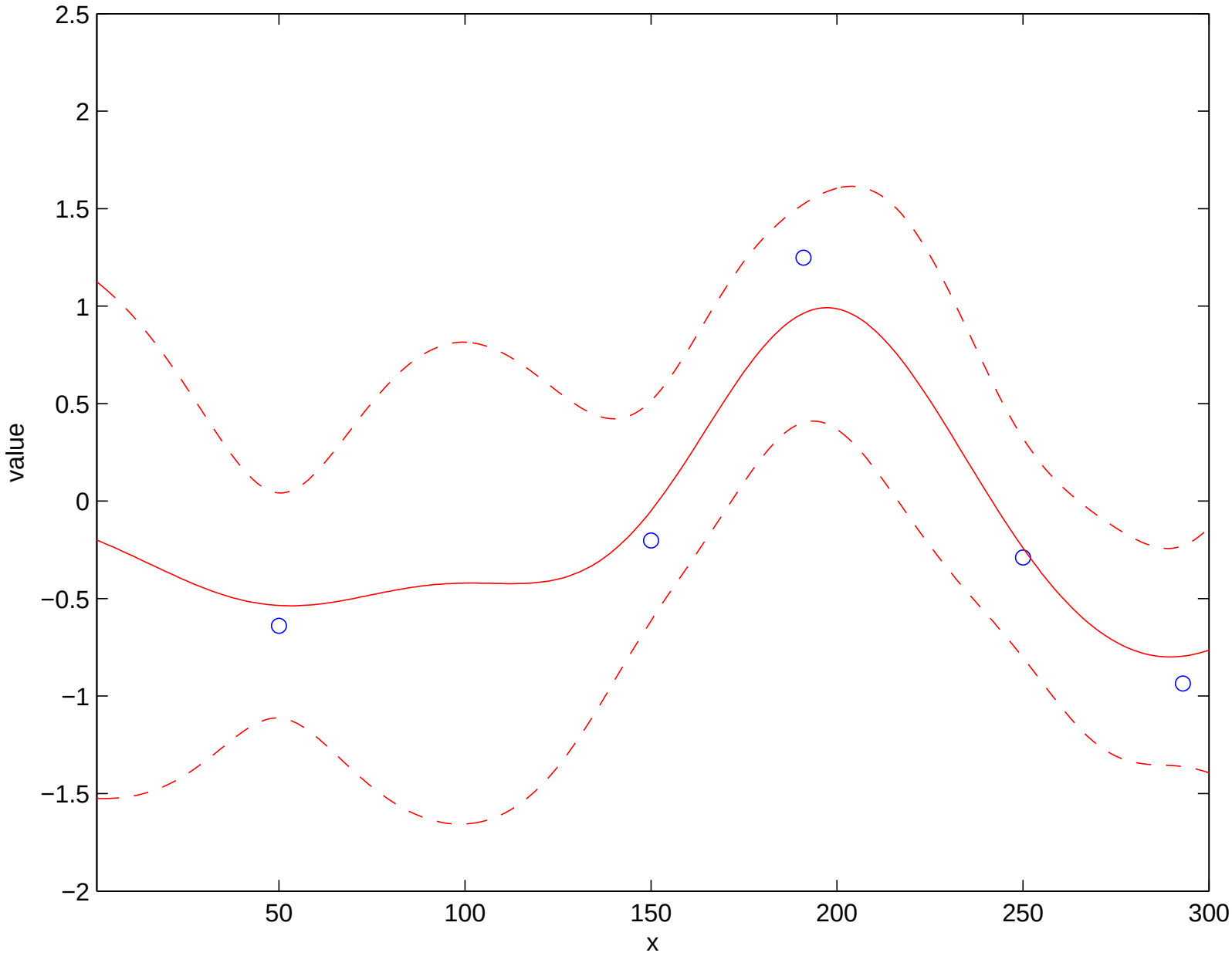
Illustrative 1D Example with Noise



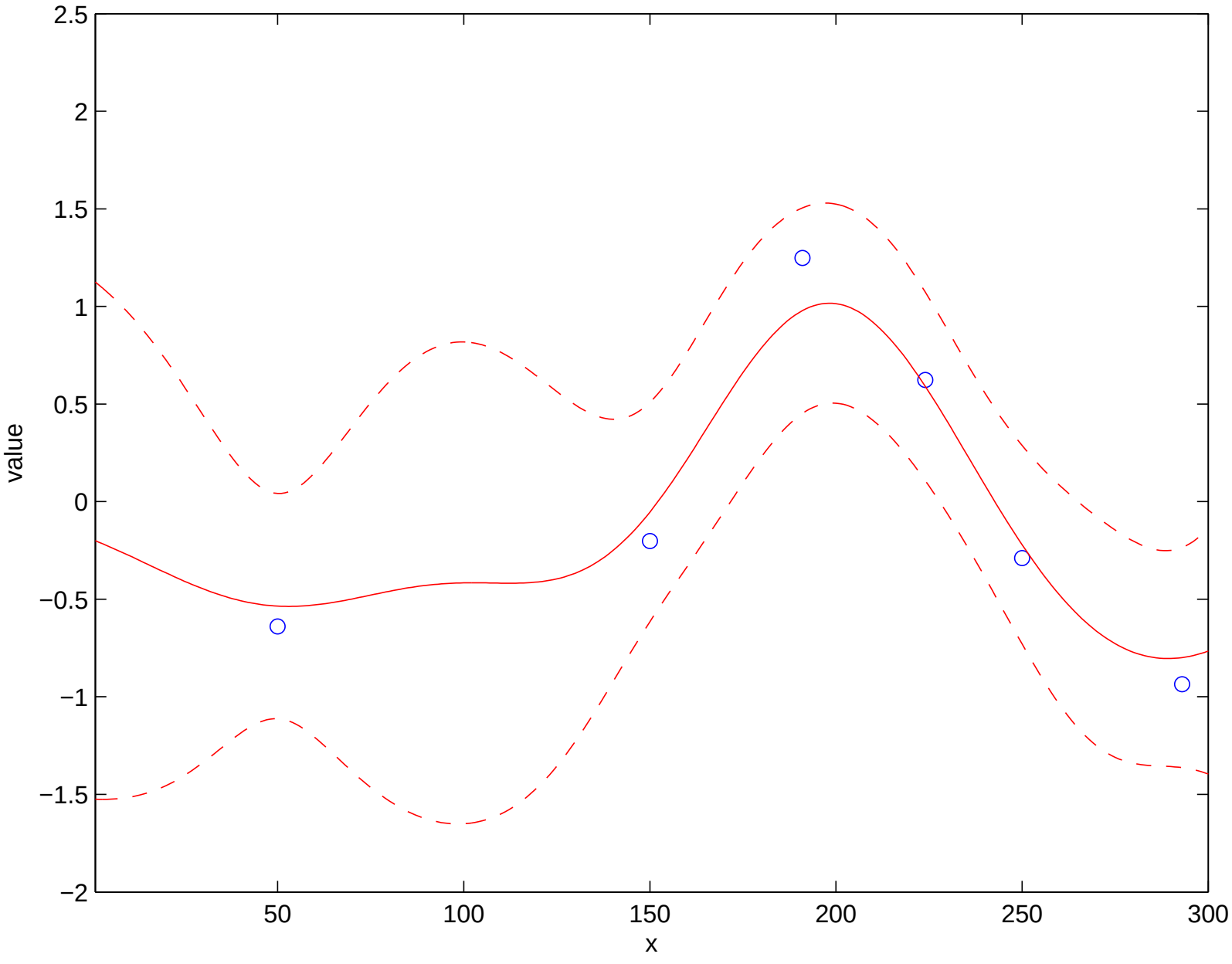
Illustrative 1D Example with Noise



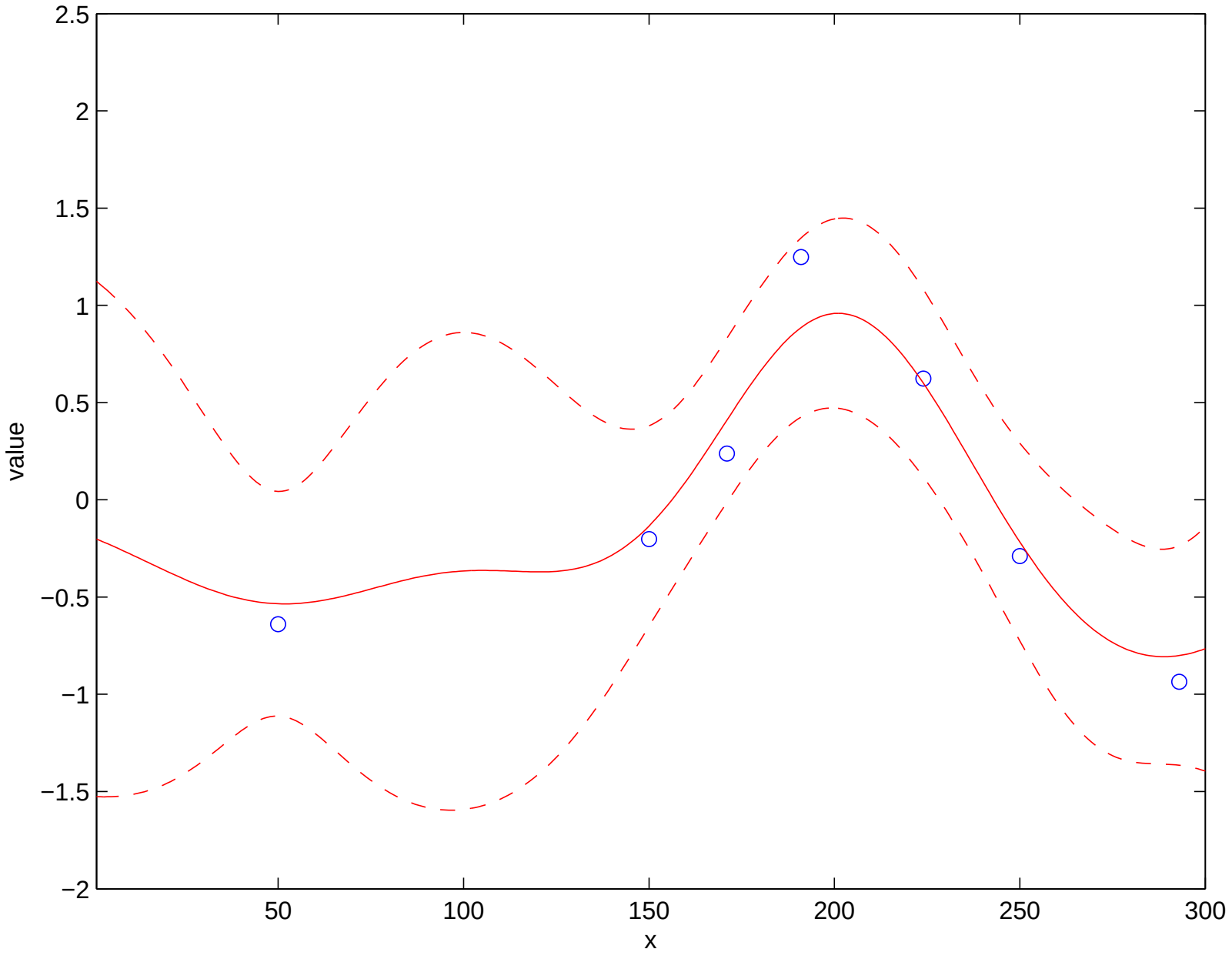
Illustrative 1D Example with Noise



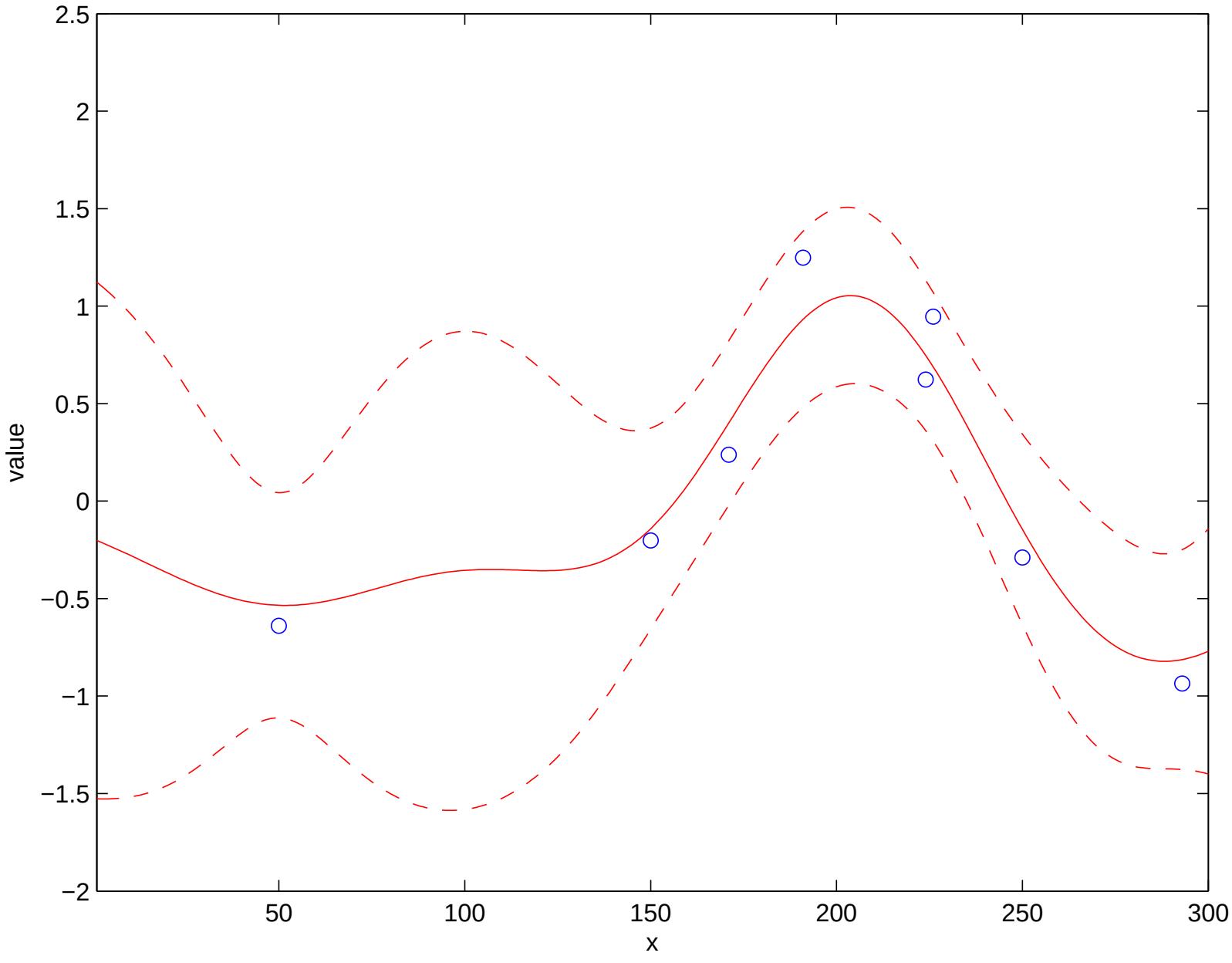
Illustrative 1D Example with Noise



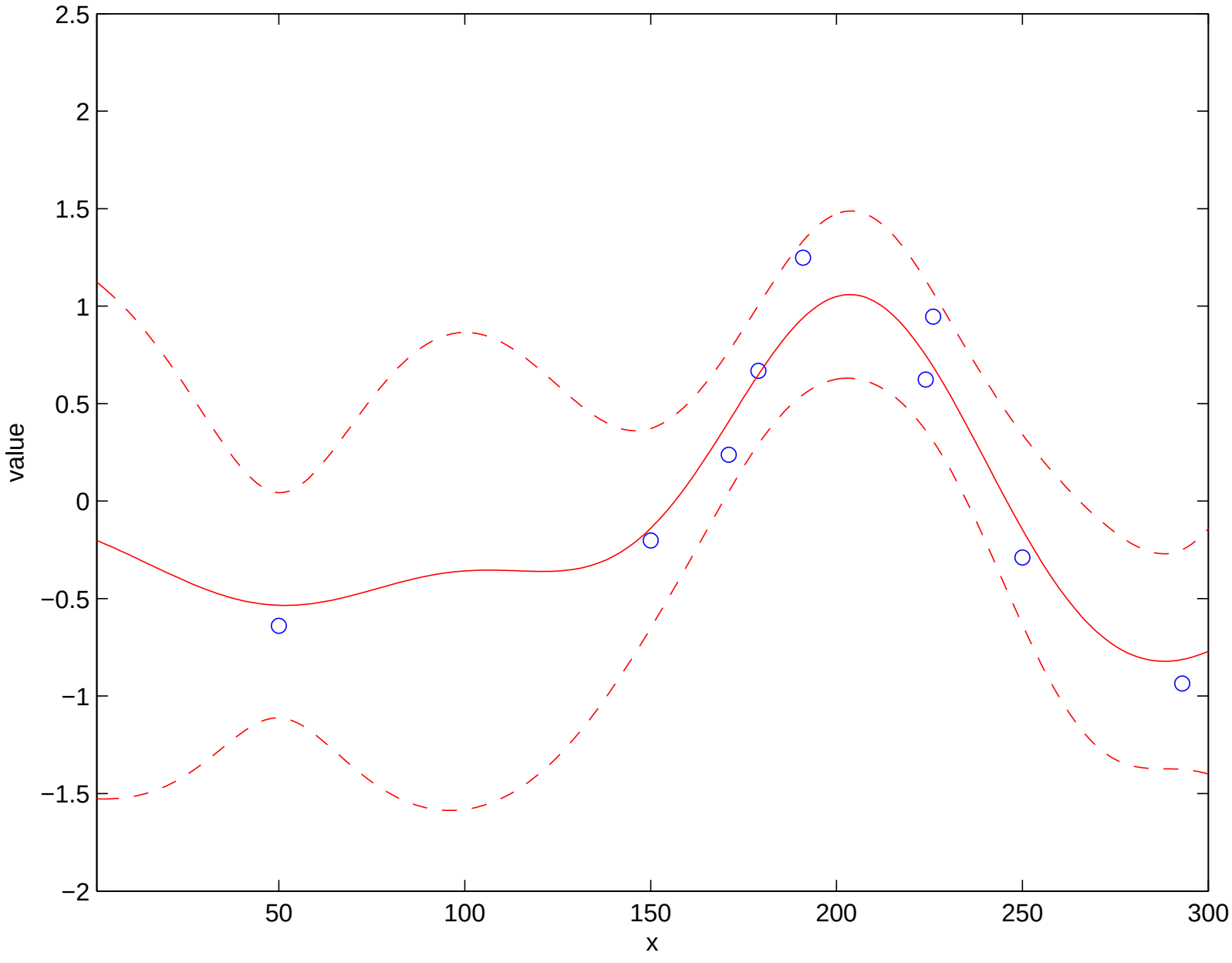
Illustrative 1D Example with Noise



Illustrative 1D Example with Noise



Illustrative 1D Example with Noise



KGCB Can be Generalized to Allow Noise

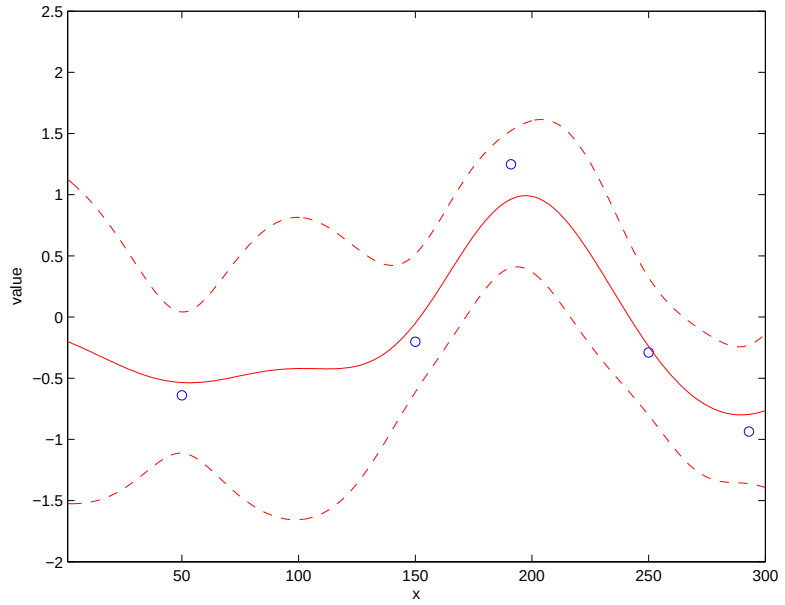
- When there is noise, the definition of the KG factor remains the same.

$$\mathrm{KG}_n(x) = \mathbb{E}_n \left[\mu^*_{n+1} - \mu^*_n \mid x_{n+1} = x \right].$$

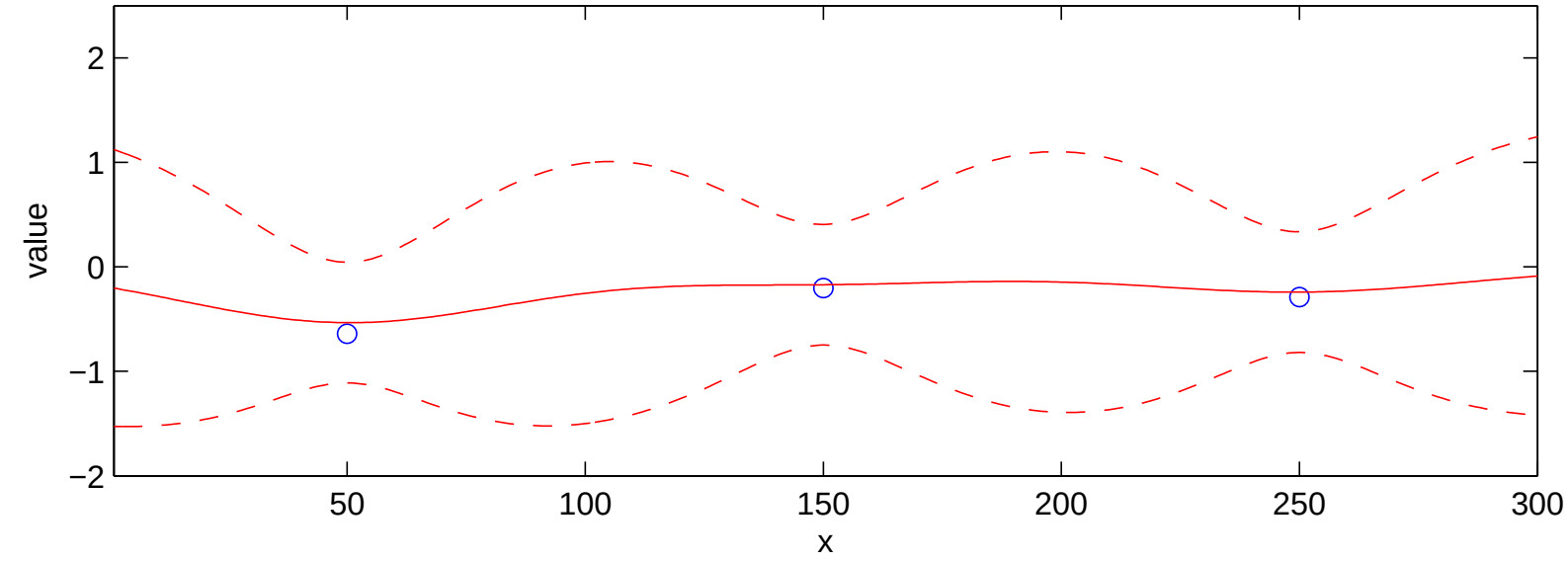
- The KGCB policy still measures at the point with the largest KG factor.

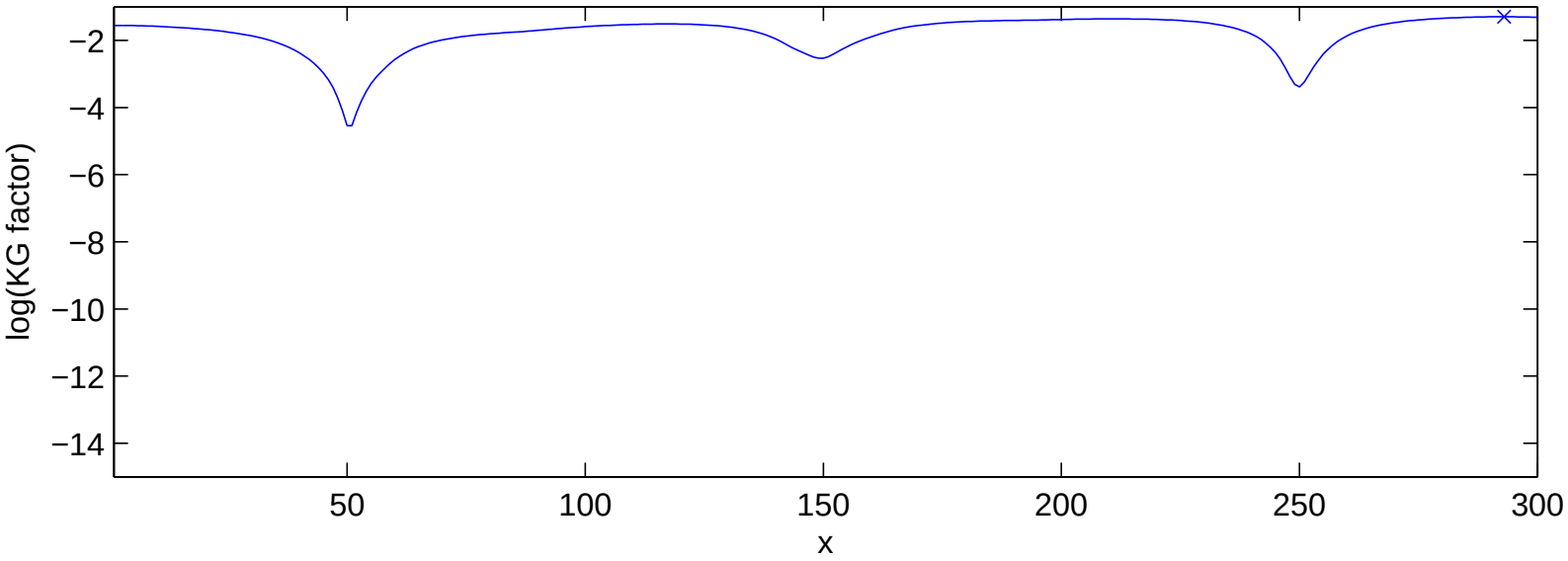
$$x_{n+1} \in \arg\max_x \mathrm{KG}_n(x).$$

- All that changes is that the estimate $\mu_n(x)$ incorporates noise.

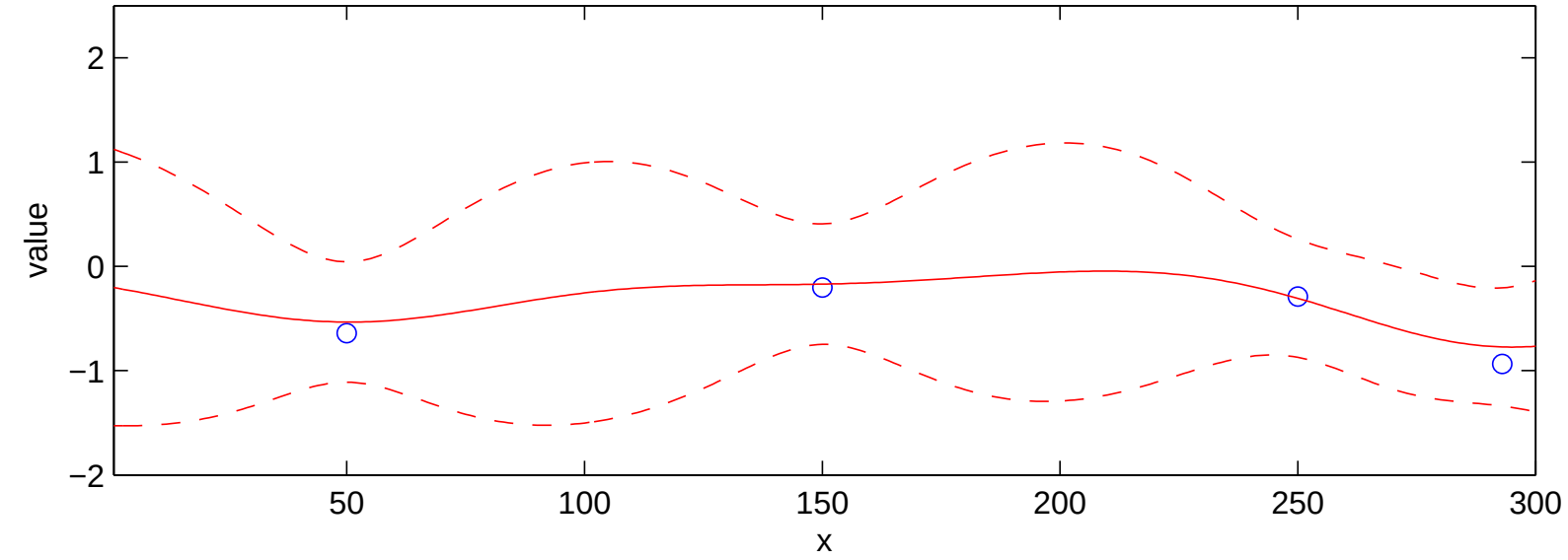


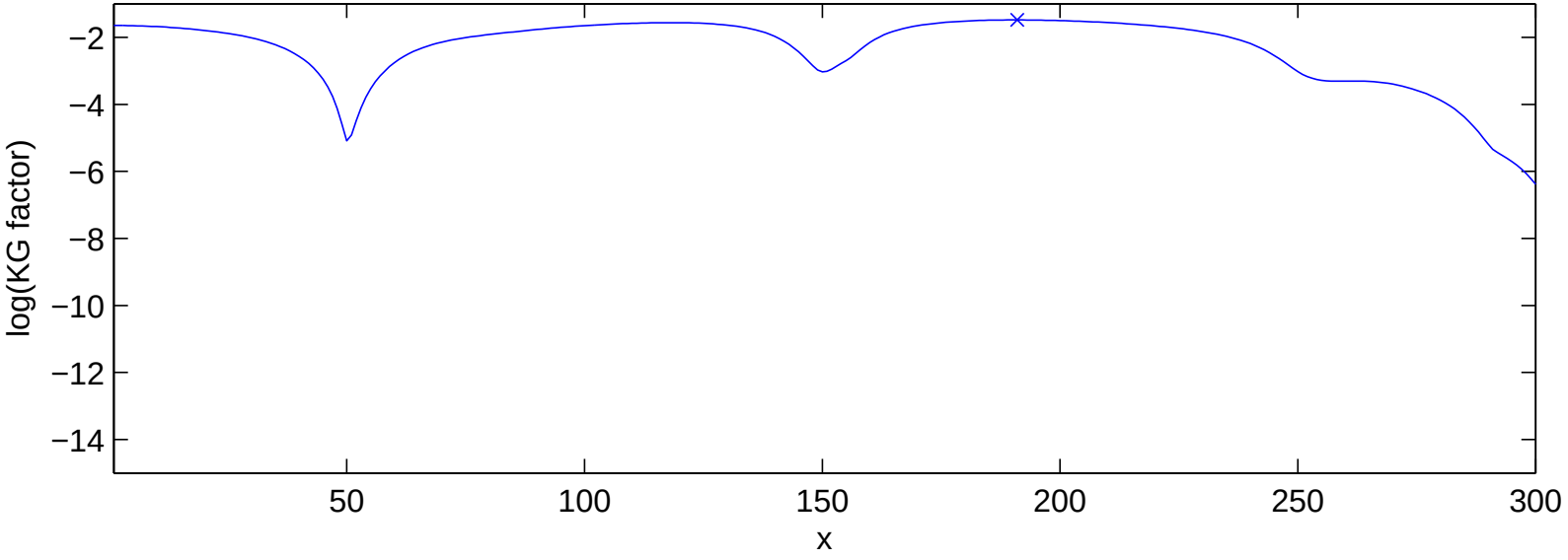
Illustrative 1D Example with Noise (KGCB)



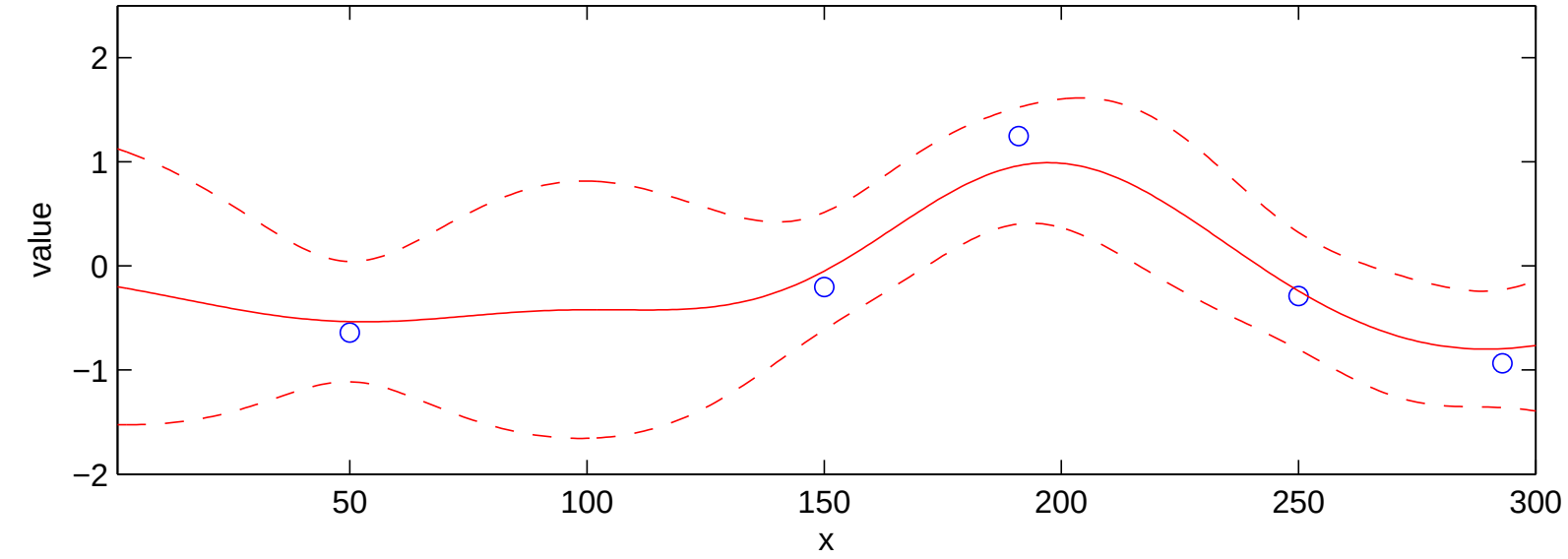


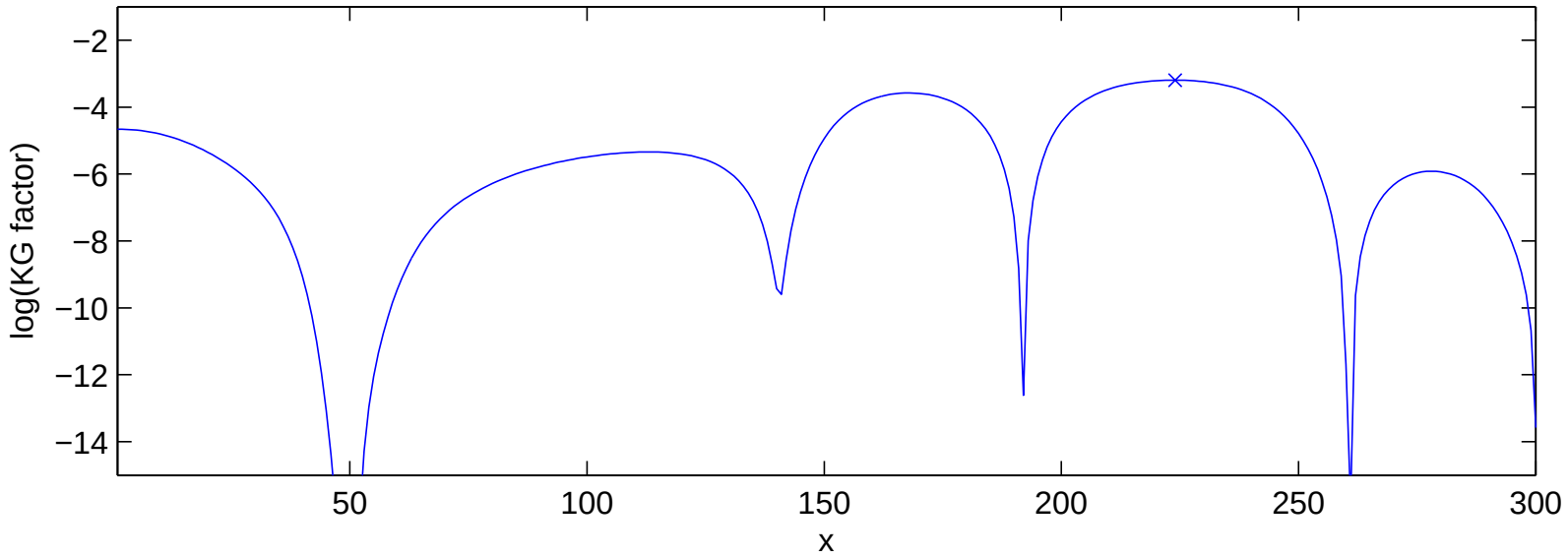
Illustrative 1D Example with Noise (KGCB)



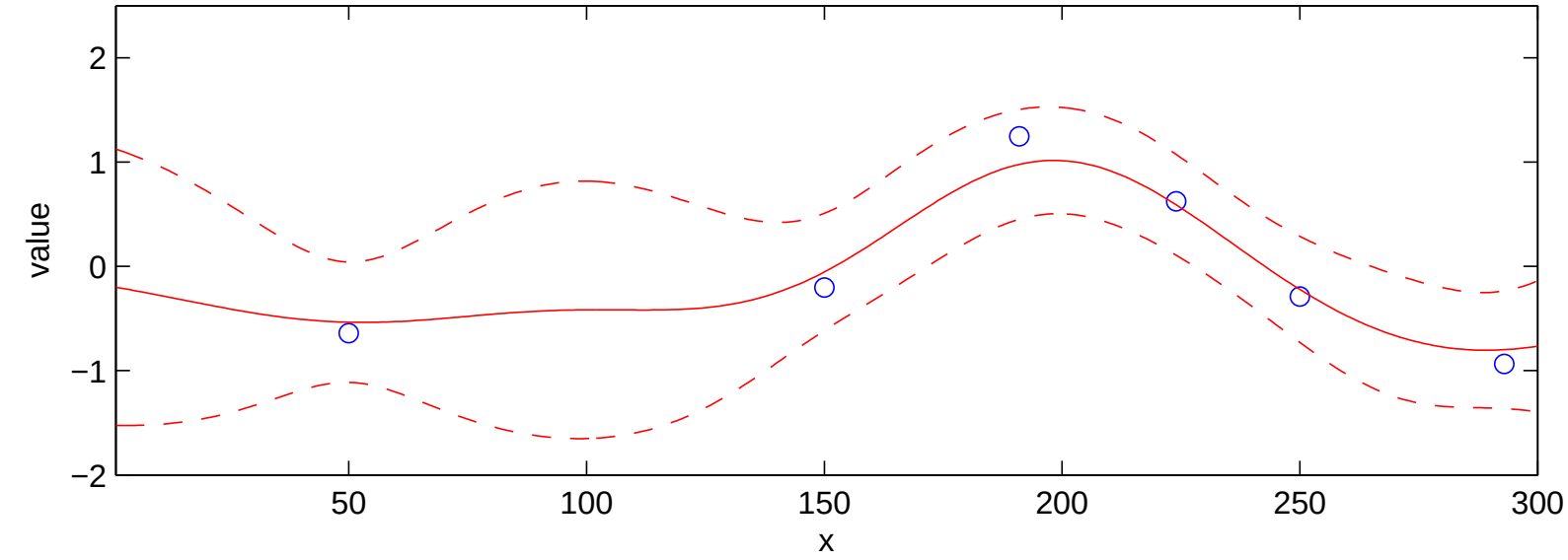


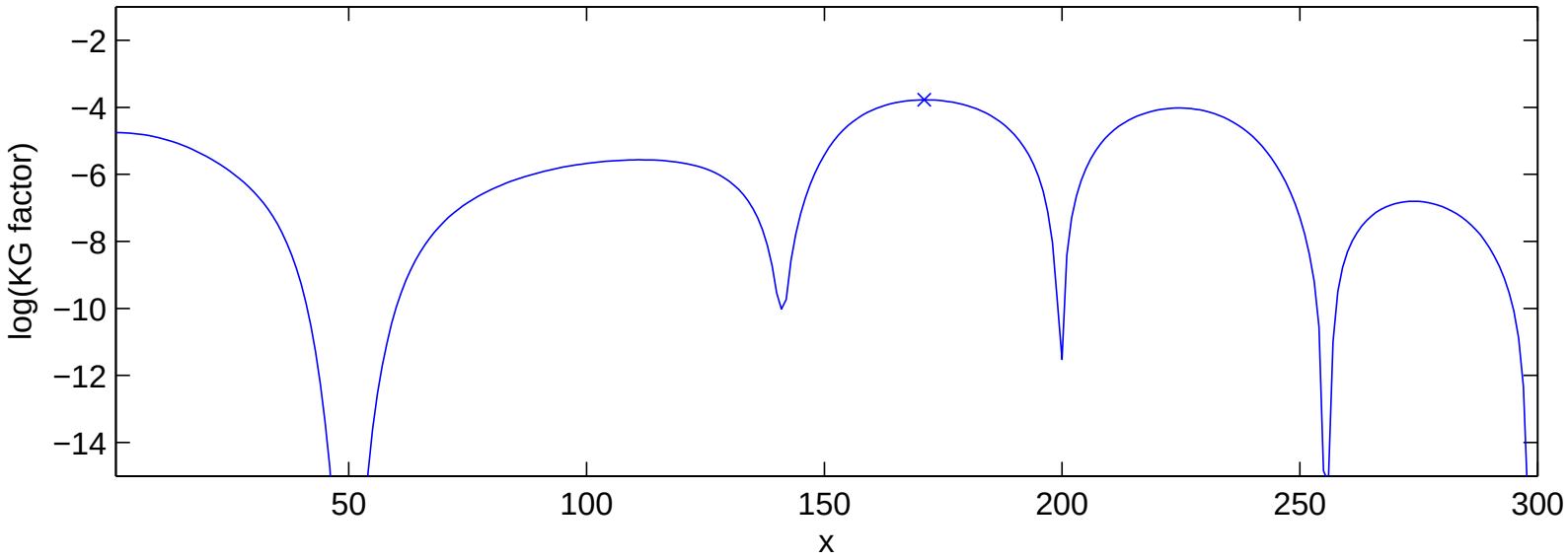
Illustrative 1D Example with Noise (KGCB)



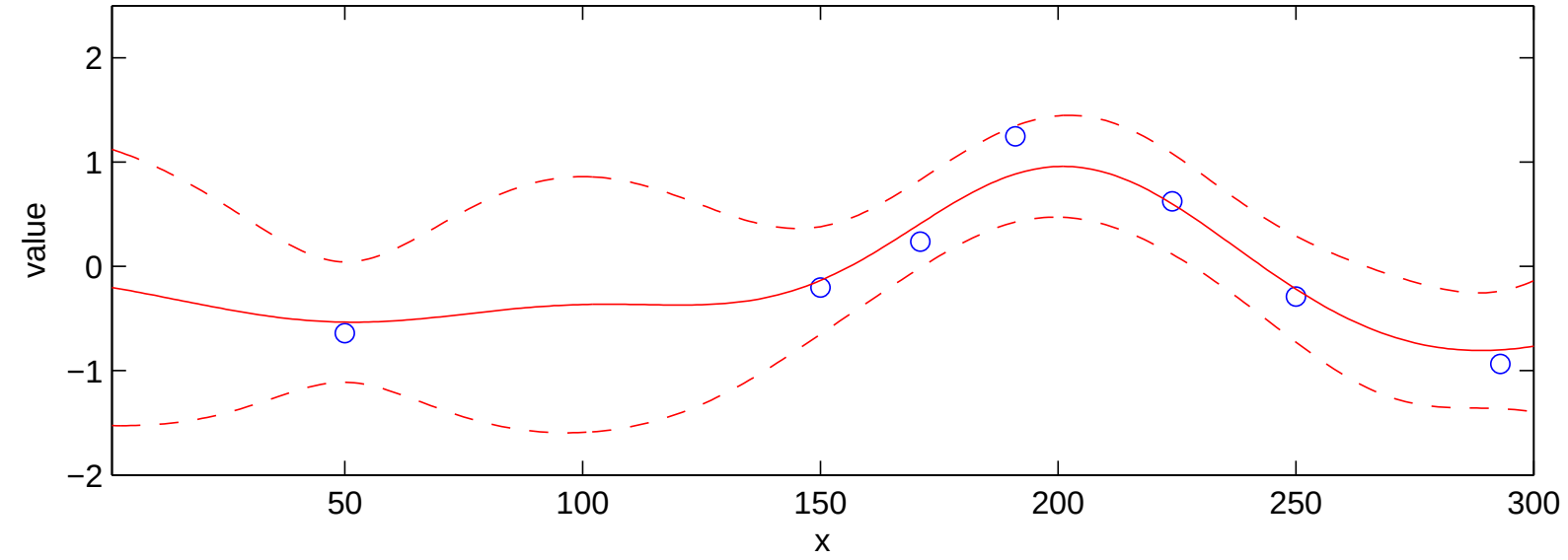


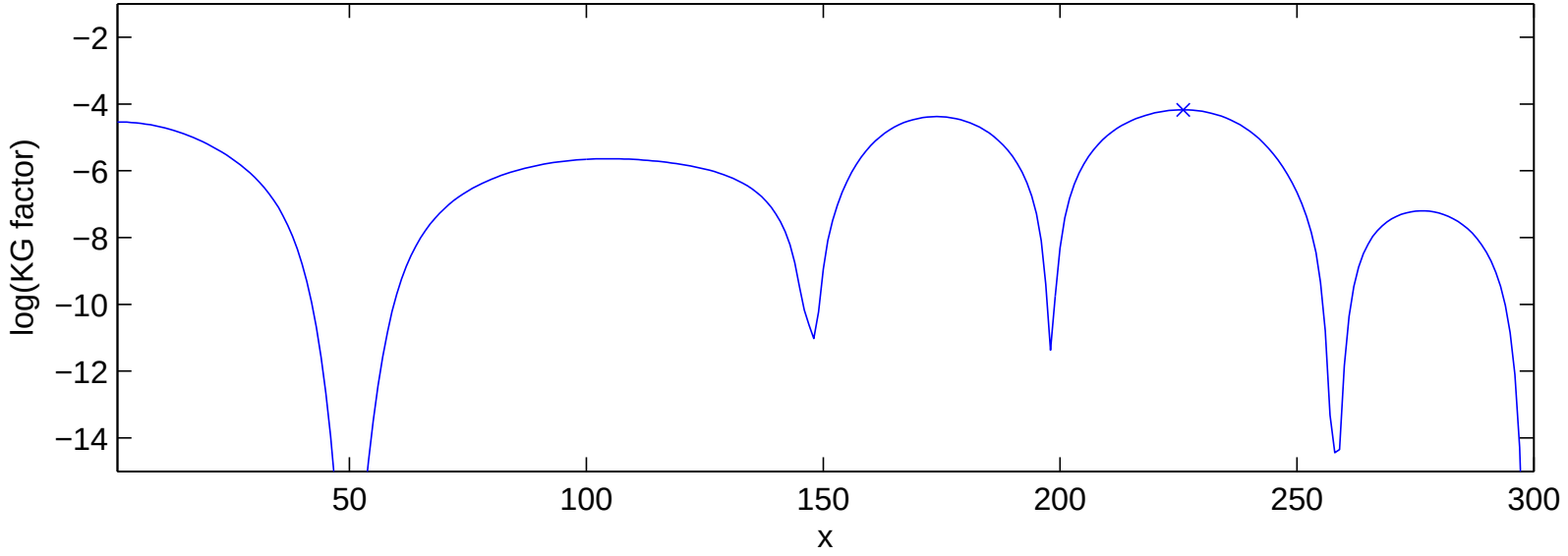
Illustrative 1D Example with Noise (KGCB)



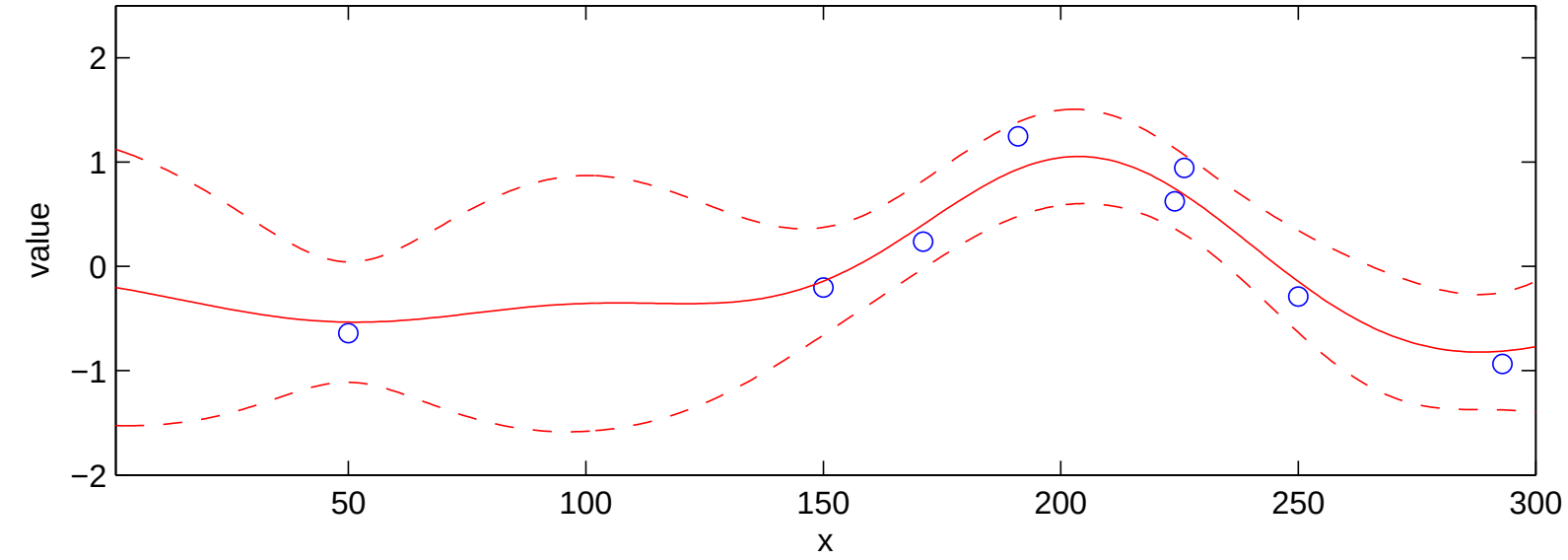


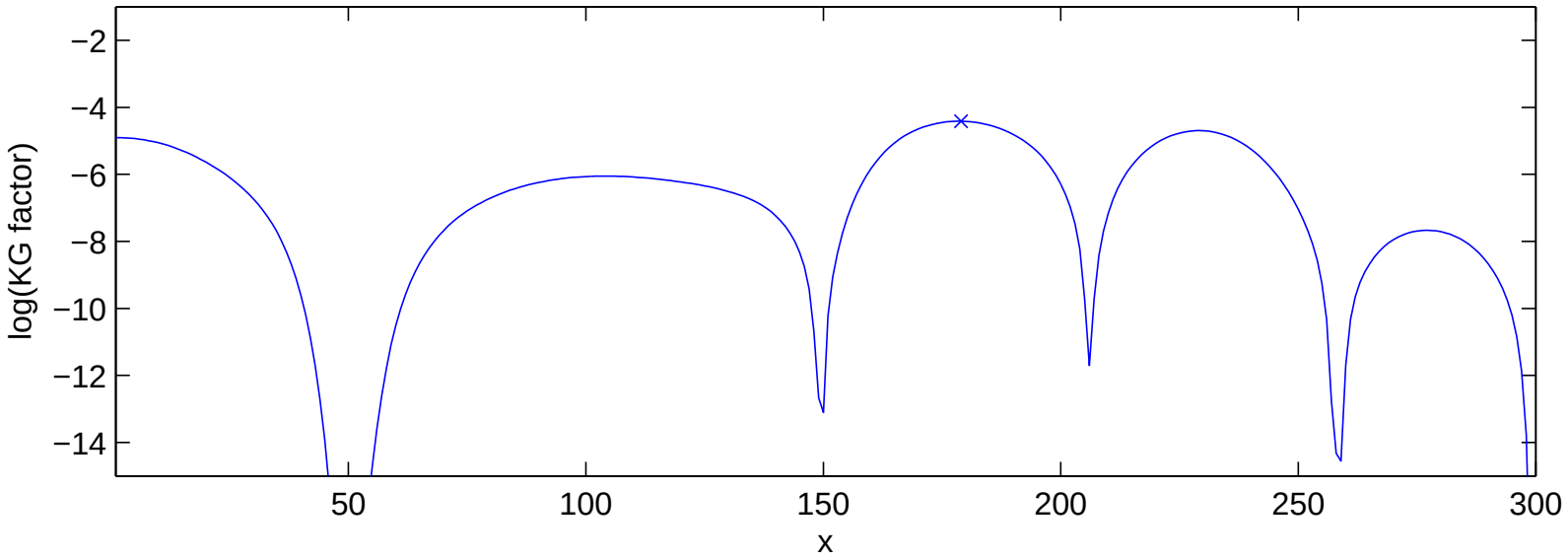
Illustrative 1D Example with Noise (KGCB)



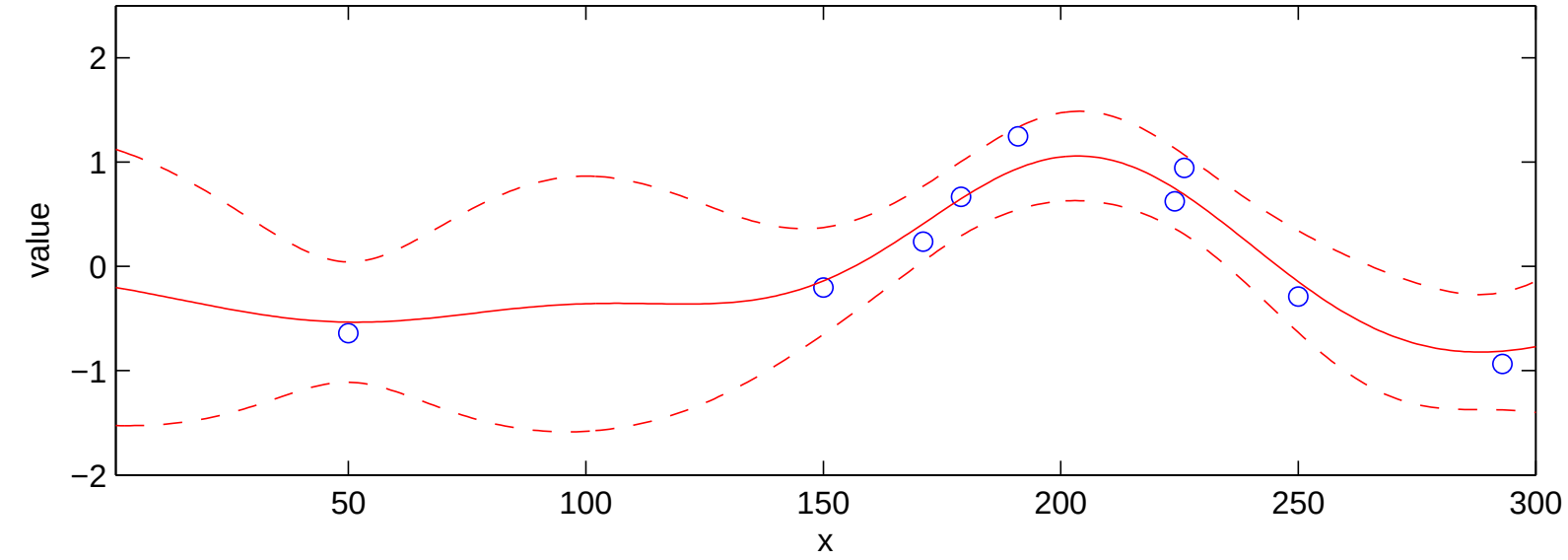


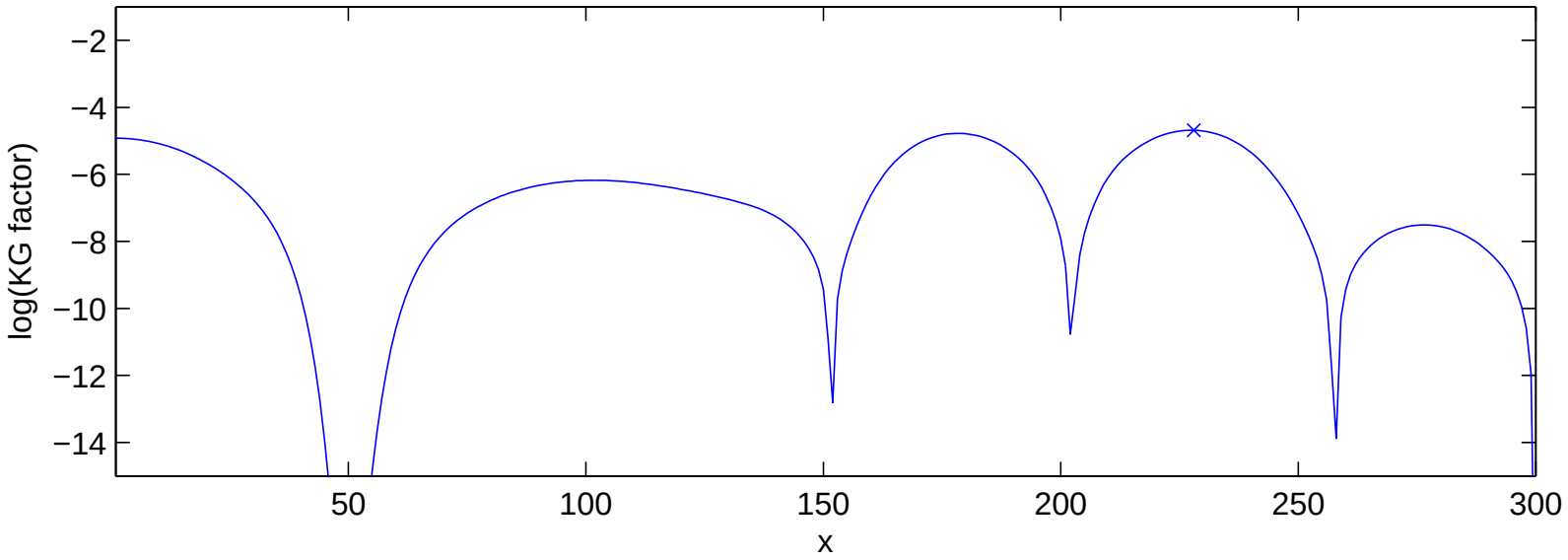
Illustrative 1D Example with Noise (KGCB)





Illustrative 1D Example with Noise (KGCB)





There are Many Other BGO Methods

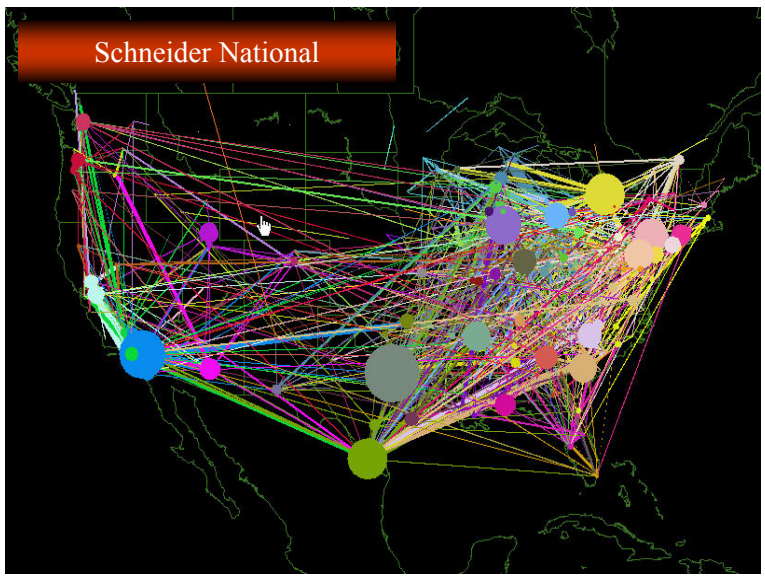
In the interests of time I will not talk about the other BGO methods: [Kushner, 1964, Mockus et al., 1978, Stuckman, 1988, Mockus, 1989, Calvin and Zilinskas, 2002, Calvin and Zilinskas, 2005, Huang et al., 2006, Forrester et al., 2006, Taddy et al., 2009, Villemonteix et al., 2009, Kleijnen et al., 2011],...

Outline

Outline

Simulation Model Calibration at Schneider National

- The logistics company Schneider National uses a large simulation-based optimization model to try "what if" scenarios.
- The model has several input parameters that must be tuned to make its behavior match reality before it can be used.
- The model is tuned by hand once per year on the most recent data. Each tuning effort requires between 1 and 2 weeks.



(Joint work with Warren B. Powell and Hugo Simão, Princeton University, [Frazier et al., 2009a])

Model Parameters

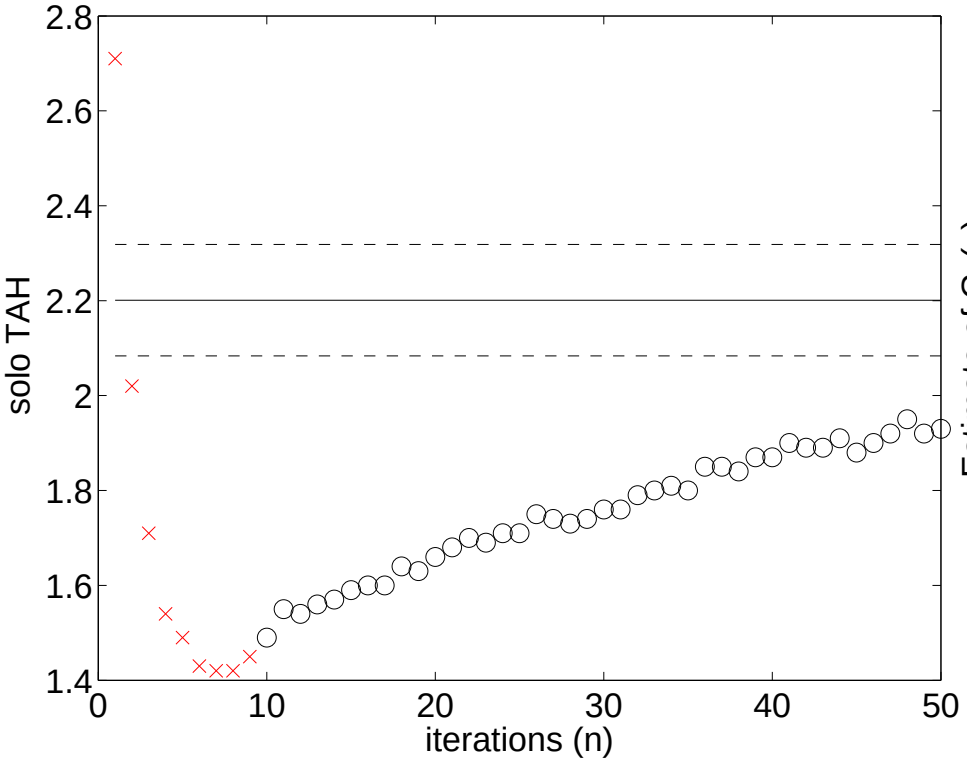
- Input parameters to the model include:
  - time-at-home bonuses.
  - "pacing" parameters describing how fast and far drivers drive per day.
  - gas prices
  - ...
- Output parameters from the model include:
  - billed miles
  - driver utilization
  - average number of trips home per driver per 4 weeks.
  - proportion of drivers without time at home over 4 weeks.
  - ...
- Some of these inputs are known (e.g., gas prices), but some are unknown (e.g. time-at-home bonuses).
- Goal: adjust the inputs to make the optimal solution found by the model match current practice.

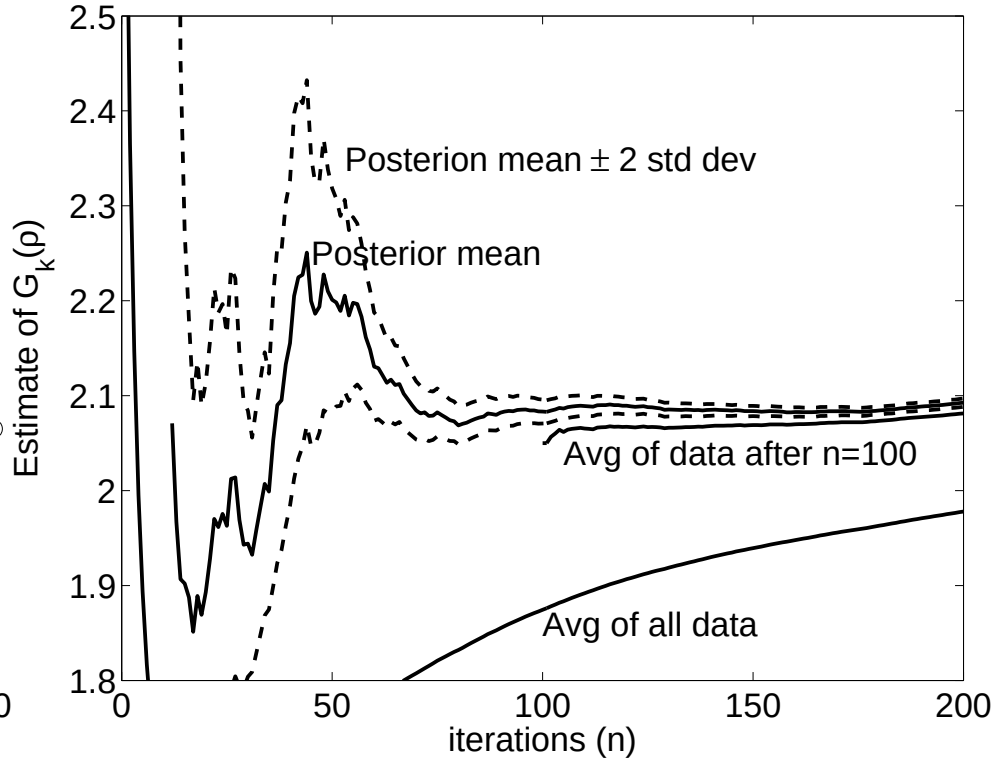
Simulation Model Calibration

- Goal: adjust the inputs to make the optimal solution found by the ADP model match current practice.
  - x is a set of inputs to the simulator.
  - $f(x)$ is how closely the simulator output matches history.
- Running the simulator for one set of bonuses takes 3 days, making calibration difficult.
- The model may be run for shorter periods of time, e.g. 12 hours, to obtain noisy output estimates.

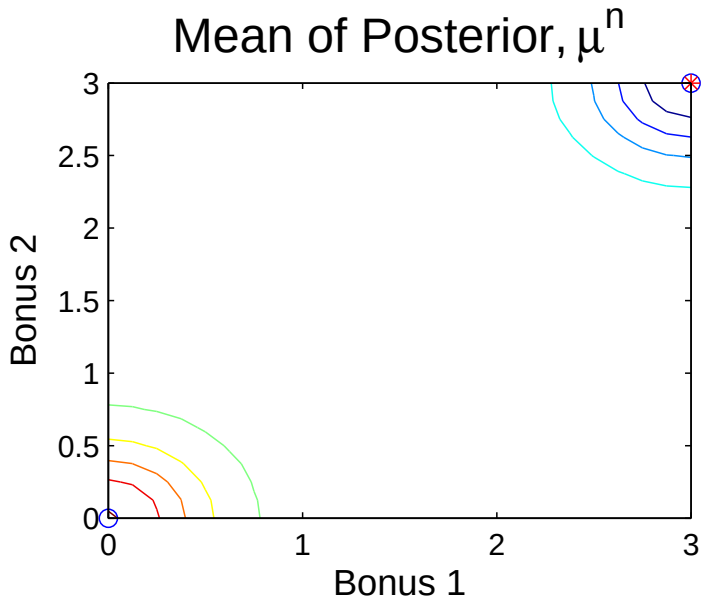
BGO is Flexible Enough to Handle Non-stationary Output

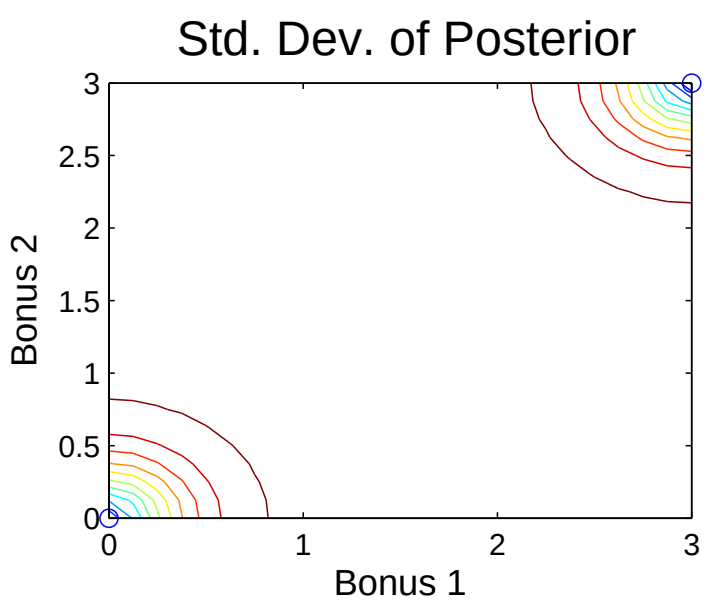
- The output of the simulator is non-stationary.
- Running the simulator to convergence takes too long (3 days).
- With just 12 hours of samples, we can use Bayesian statistics to get a noisy estimate of where the path is going.

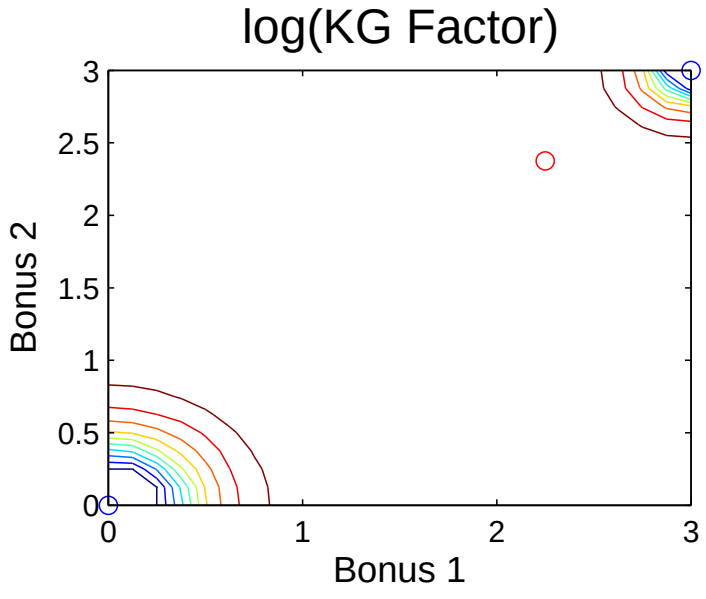


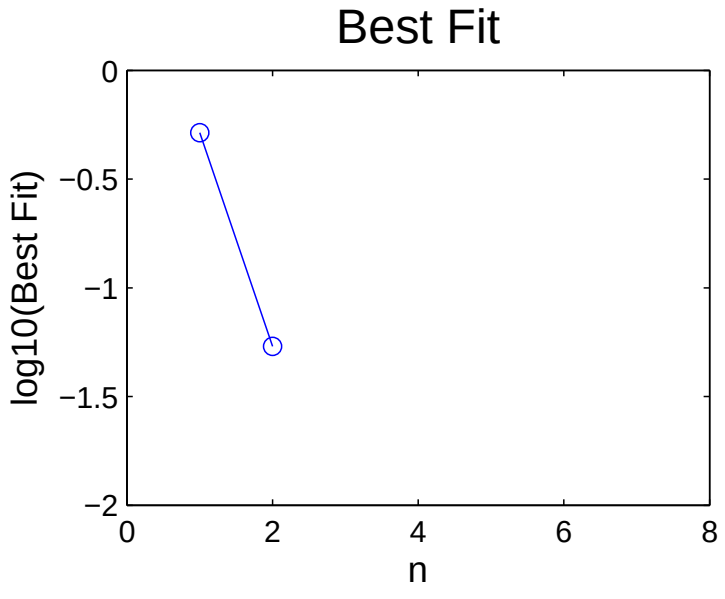


Simulation Model Calibration Results

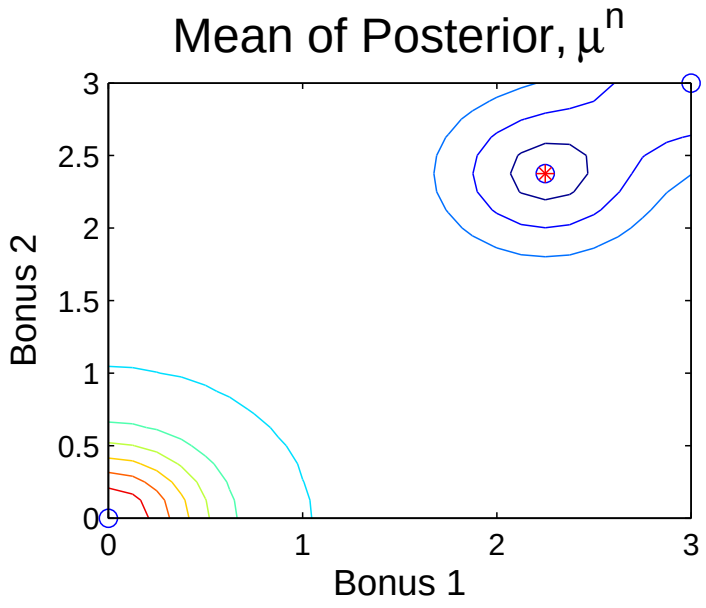


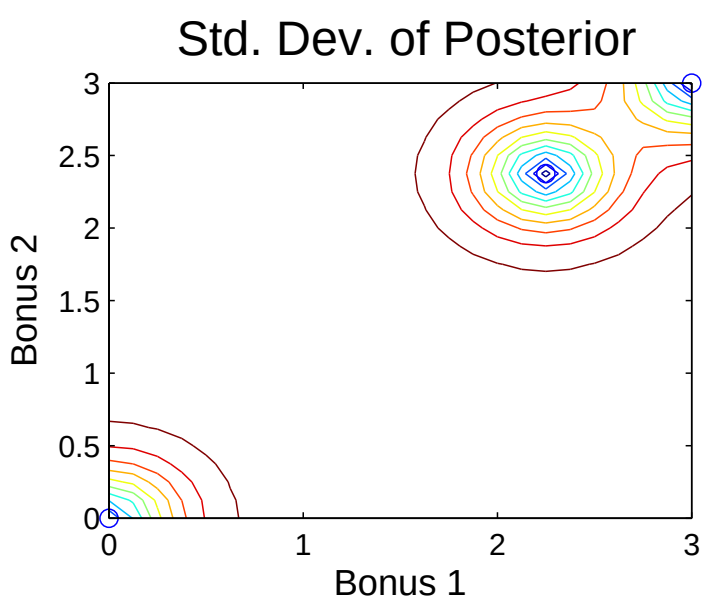


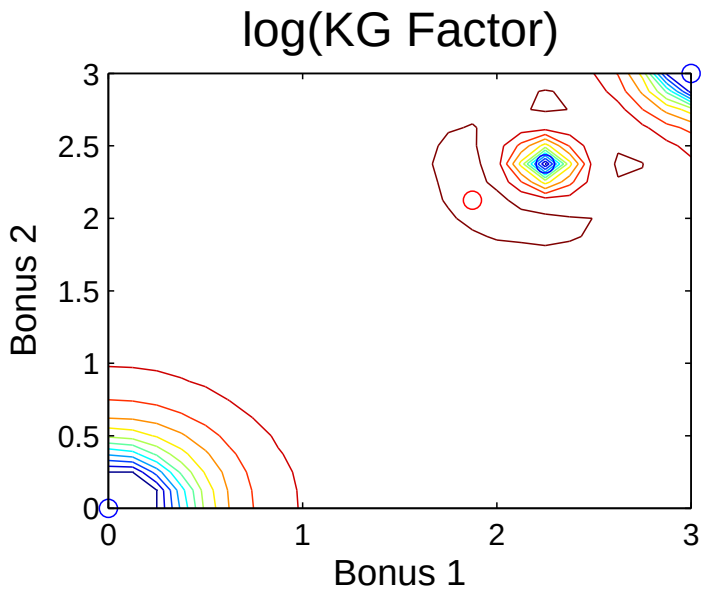


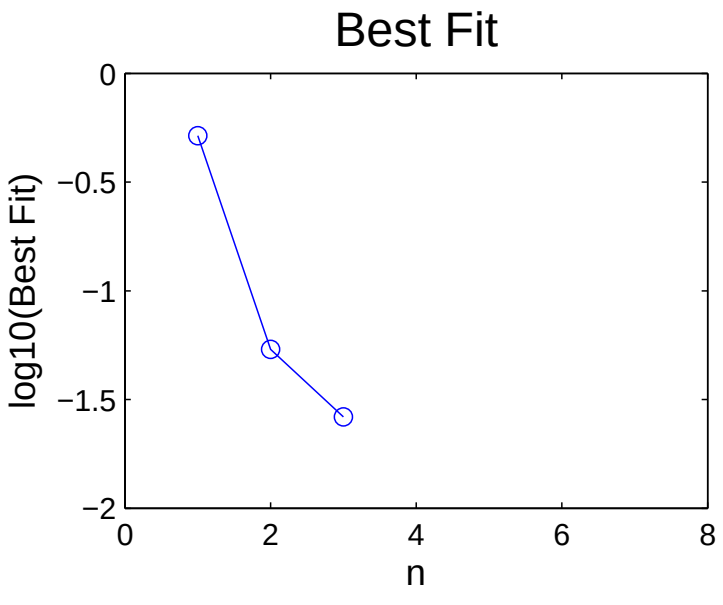


Simulation Model Calibration Results

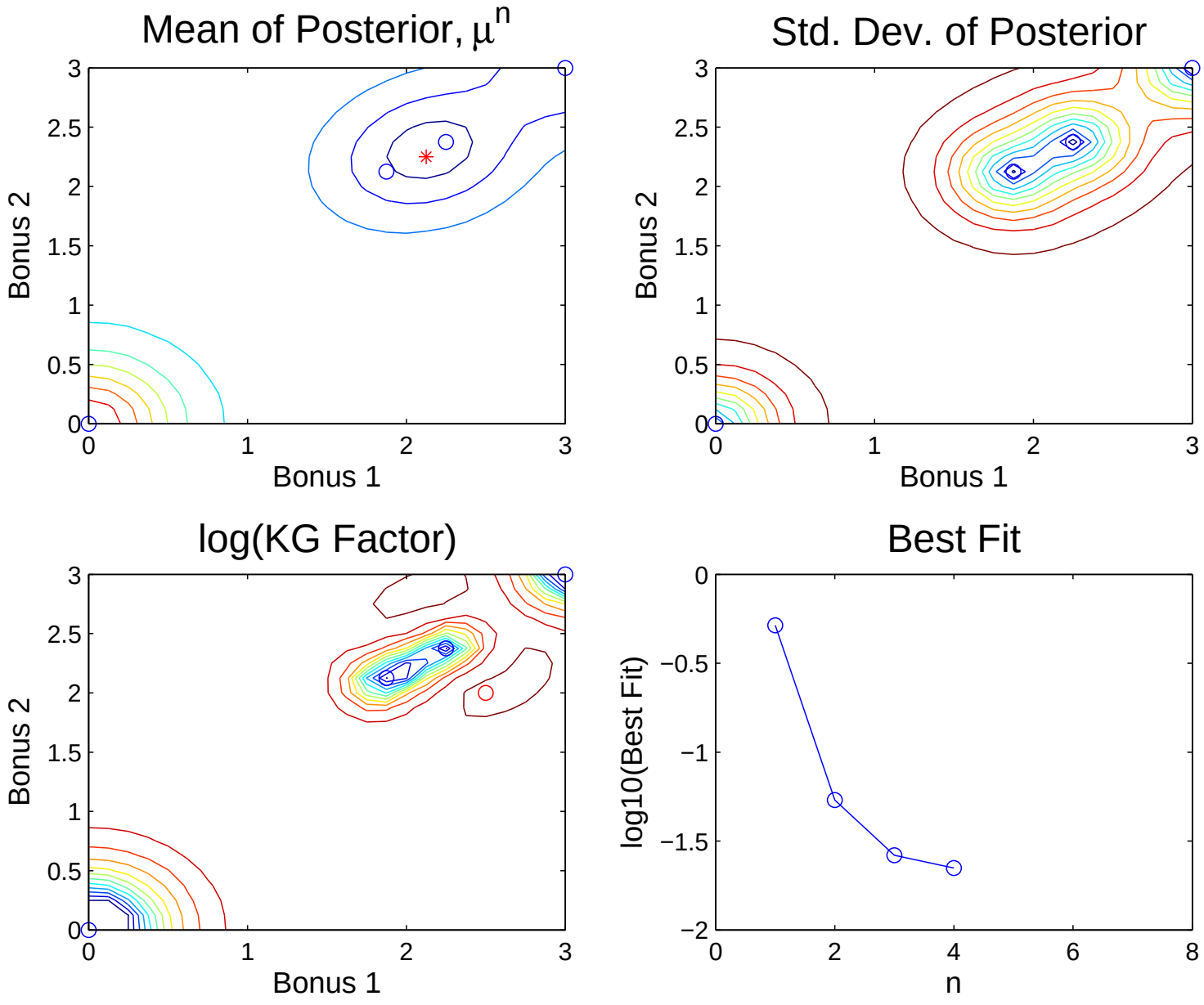




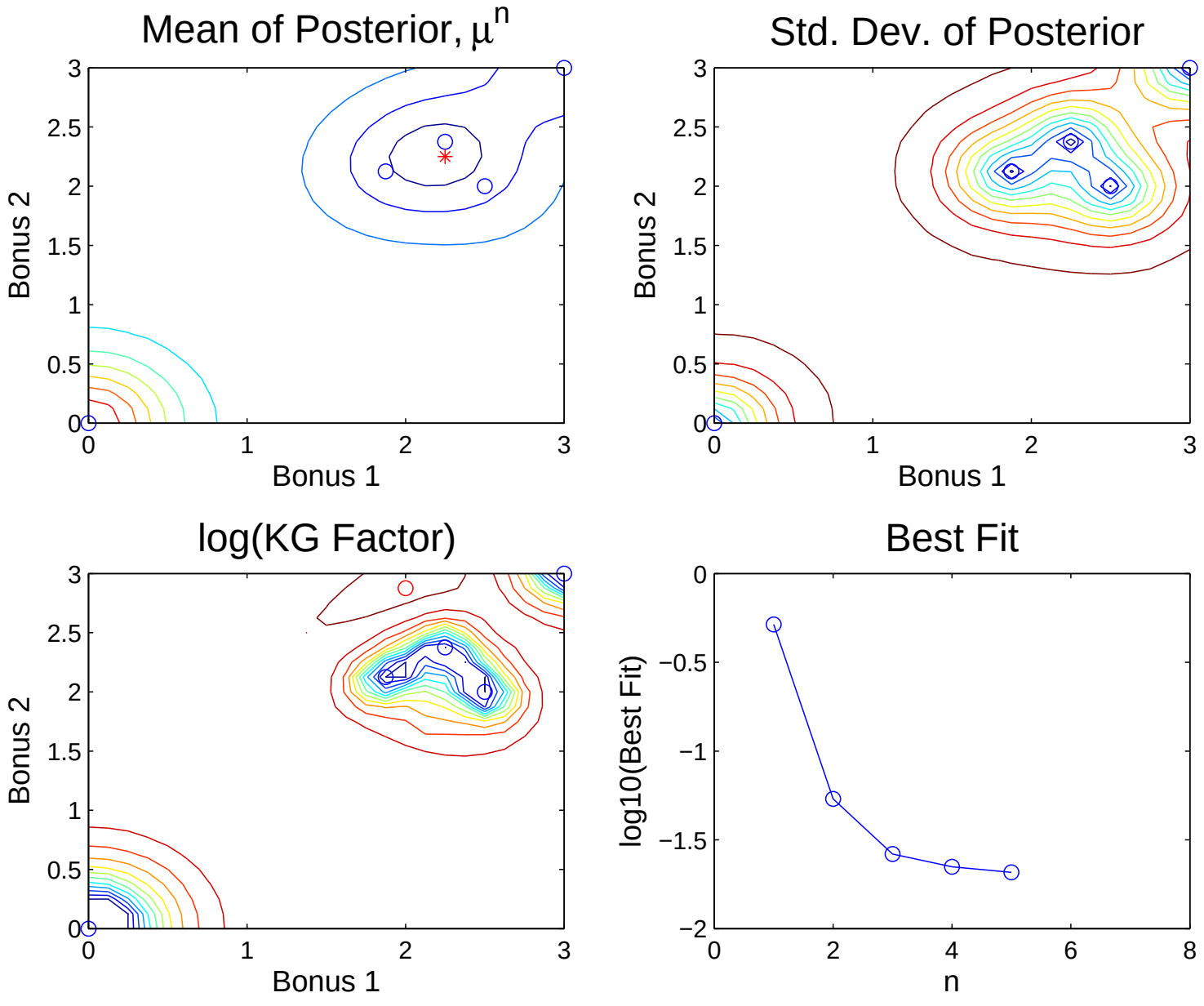




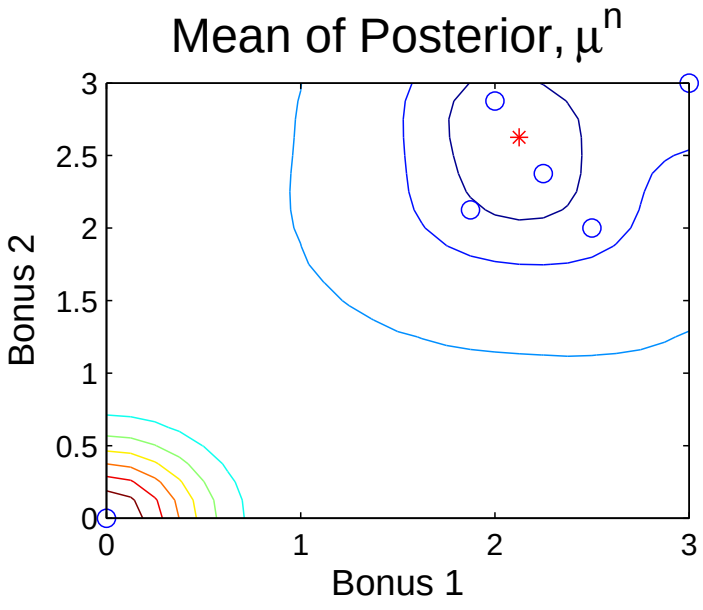
Simulation Model Calibration Results

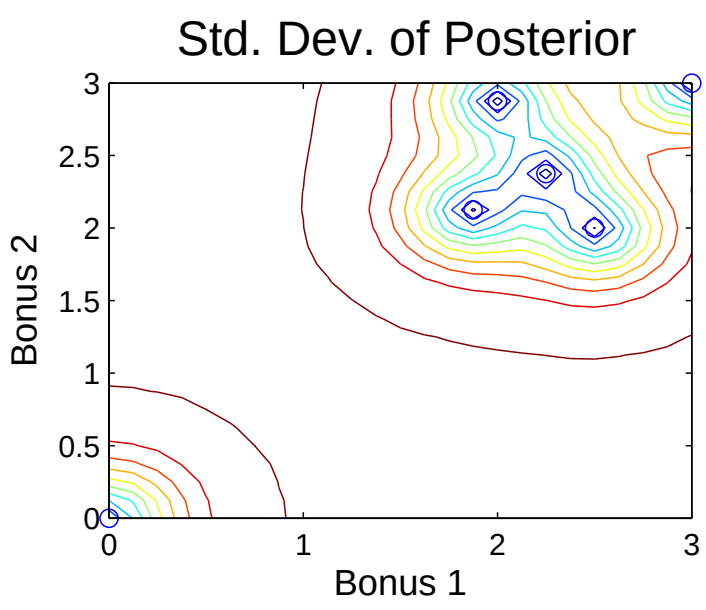


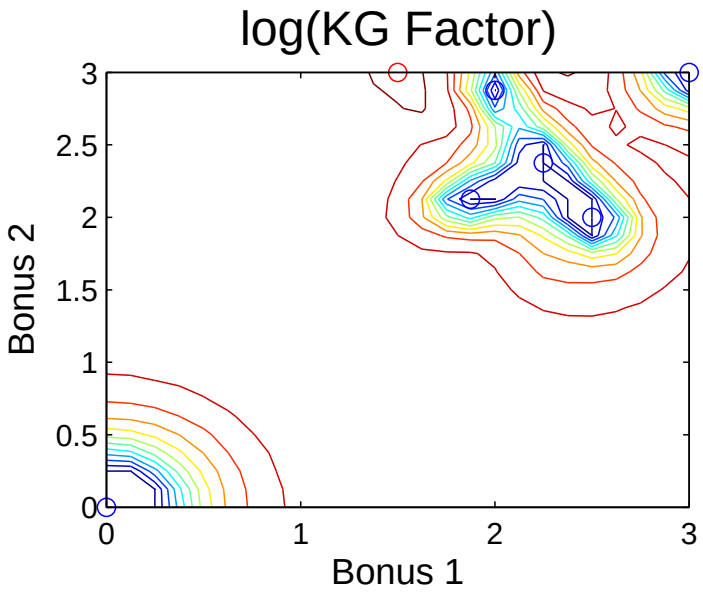
Simulation Model Calibration Results

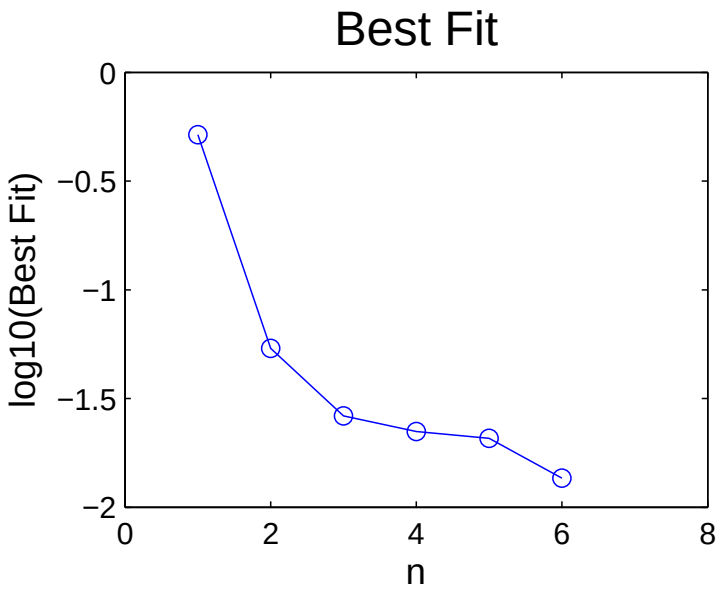


Simulation Model Calibration Results

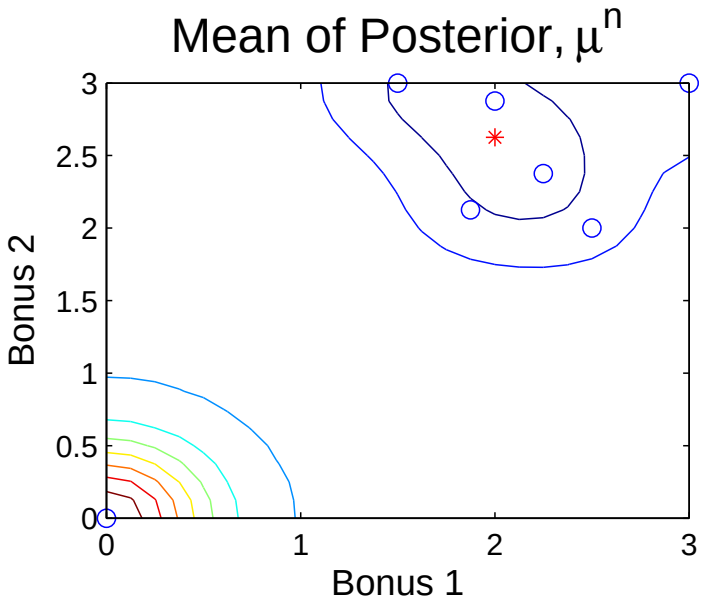


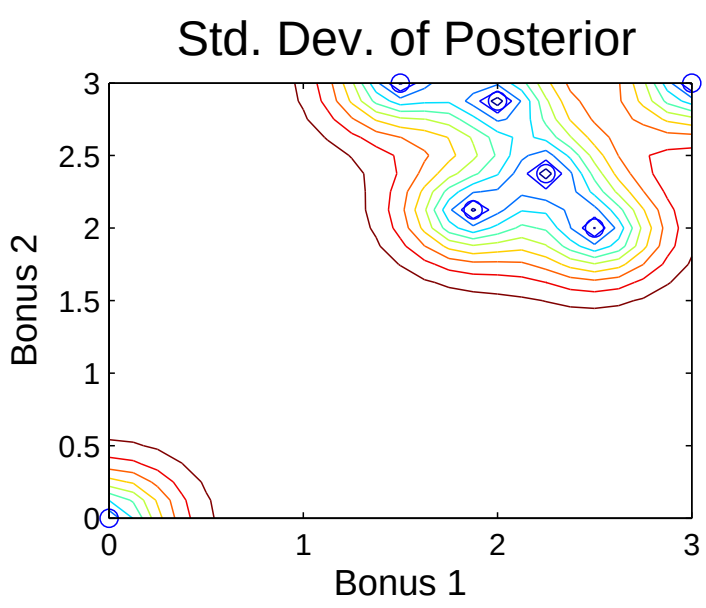


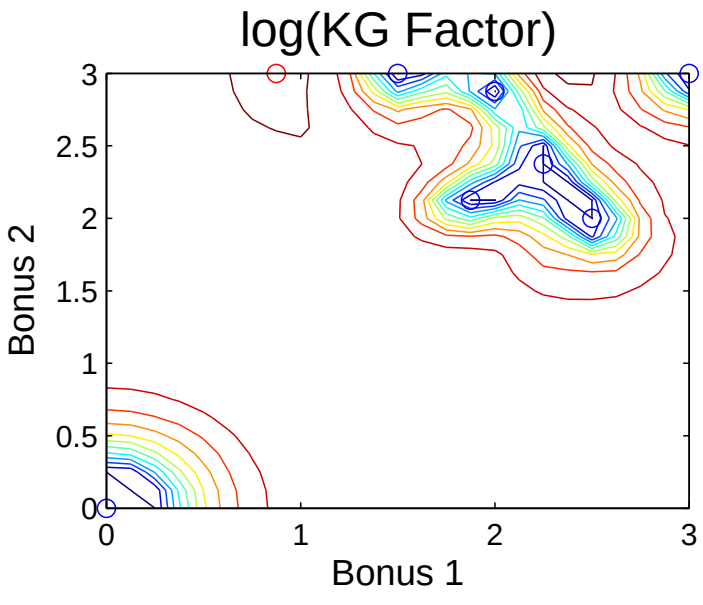


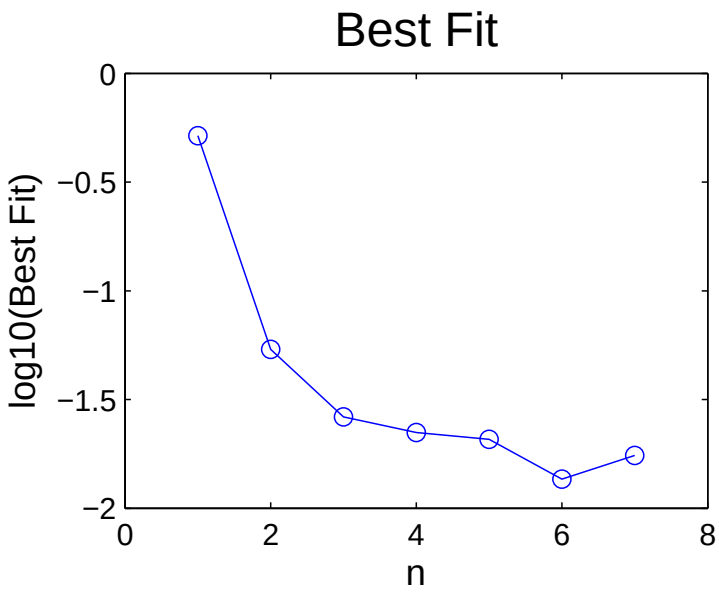


Simulation Model Calibration Results









Simulation Model Calibration Results

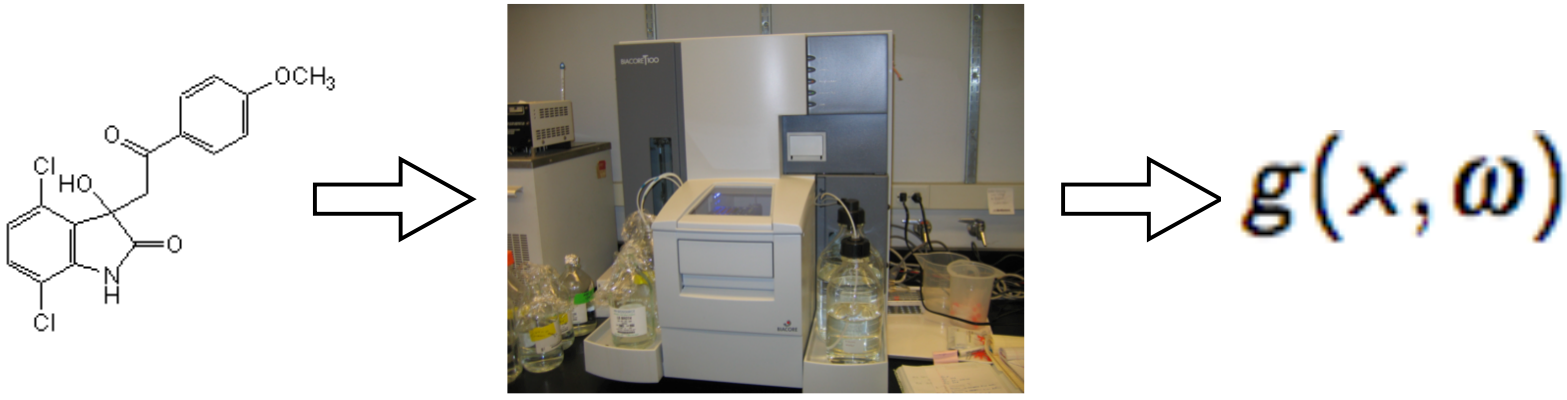
- The KG method calibrates the model in approximately 3 days, compared to $7-14$ days when tuned by hand.
- The calibration is automatic, freeing the human calibrator to do other work.
- The KG method calibrates as accurately or better than does by-hand calibration.
- Current practice uses the year's calibrated bonuses for each new "what if" scenario, but to enforce the constraint on driver at-home time it would be better to recalibrate the model for each scenario. Automatic calibration with the KG method makes this feasible.

Outline

Ewings Sarcoma is a Pediatric Bone Cancer

Long-term survival rate is $60-80\%$ for localized disease, and $\approx 20\%$ following metastisis.

Drug Development is Global Optimization

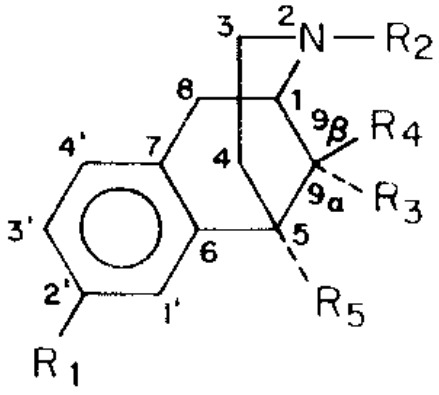


- We have a large number of chemically related small molecules, some of which might make a good drug.
- We can synthesize and test the quality of these molecules, but each molecule tested takes days of effort.
- $f(x)$ is the quality of molecule x, and $g(x, \omega)$ is the test result.
- We would like to find a good drug with a limited number of tests.

Joint work with Jeffrey Toretsky, M.D. (Georgetown), Diana Negoescu (Stanford), Warren B. Powell (Princeton), [Negoescu et al., 2011])

We Use a Gaussian Process Prior

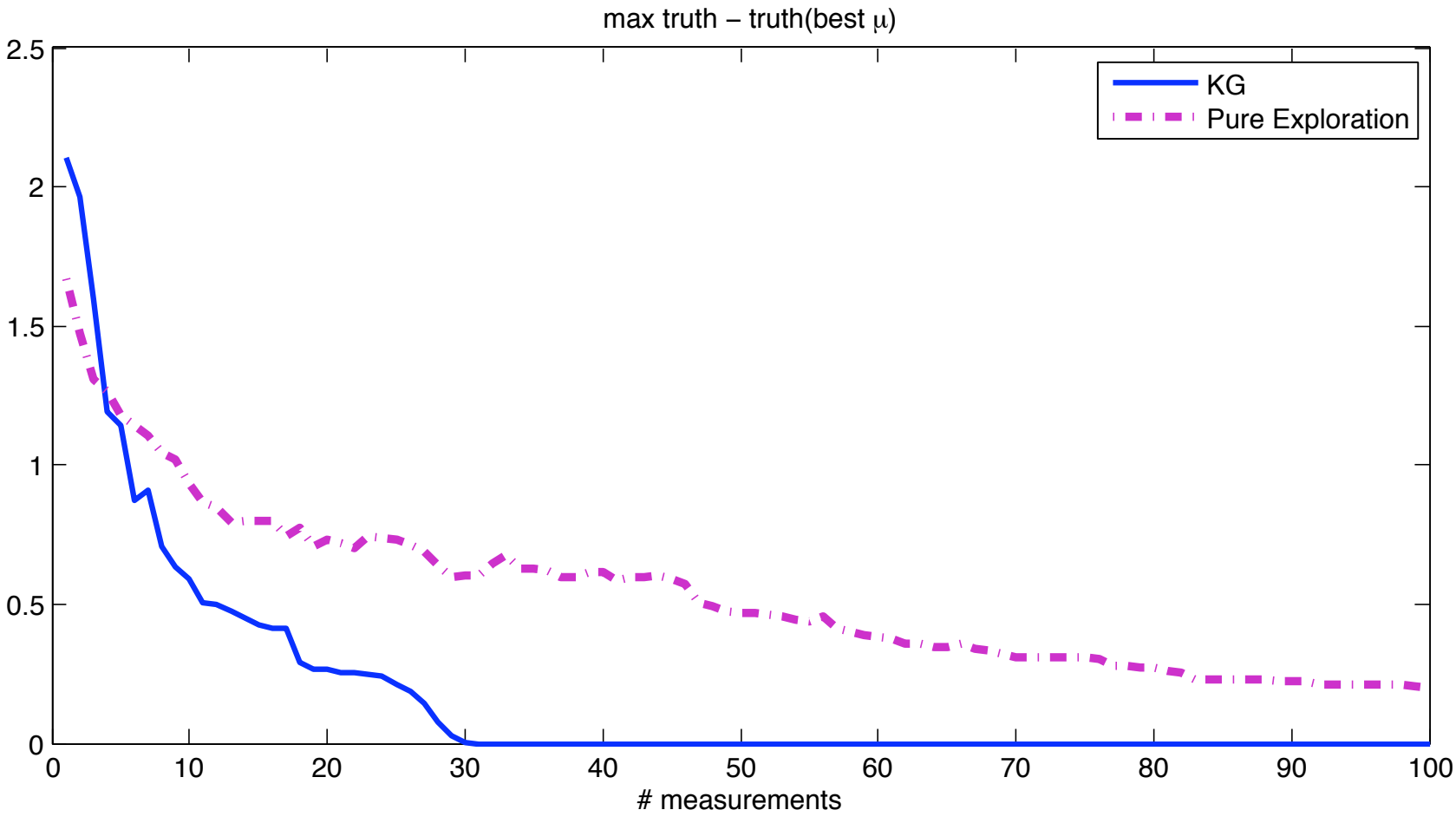
- The molecules we consider share a common skeleton, and are described by which substituents are present at each location.



Substituent
$R_1 = F$
$R_1 = OH$
$R_1 = OCOCH_3$
$R_1 = OCOCH_2CH_3$
$R_1 = OCOCH_2CH_2CH_3$
$R_1 = OCOCH_2(CH_2)_2CH_3$
$R_1 = OCH_3$
$R_1 = NO_2$
$R_1 = OCOAr(3\text{-}C_5H_4N)$
$R_1 = Cl$

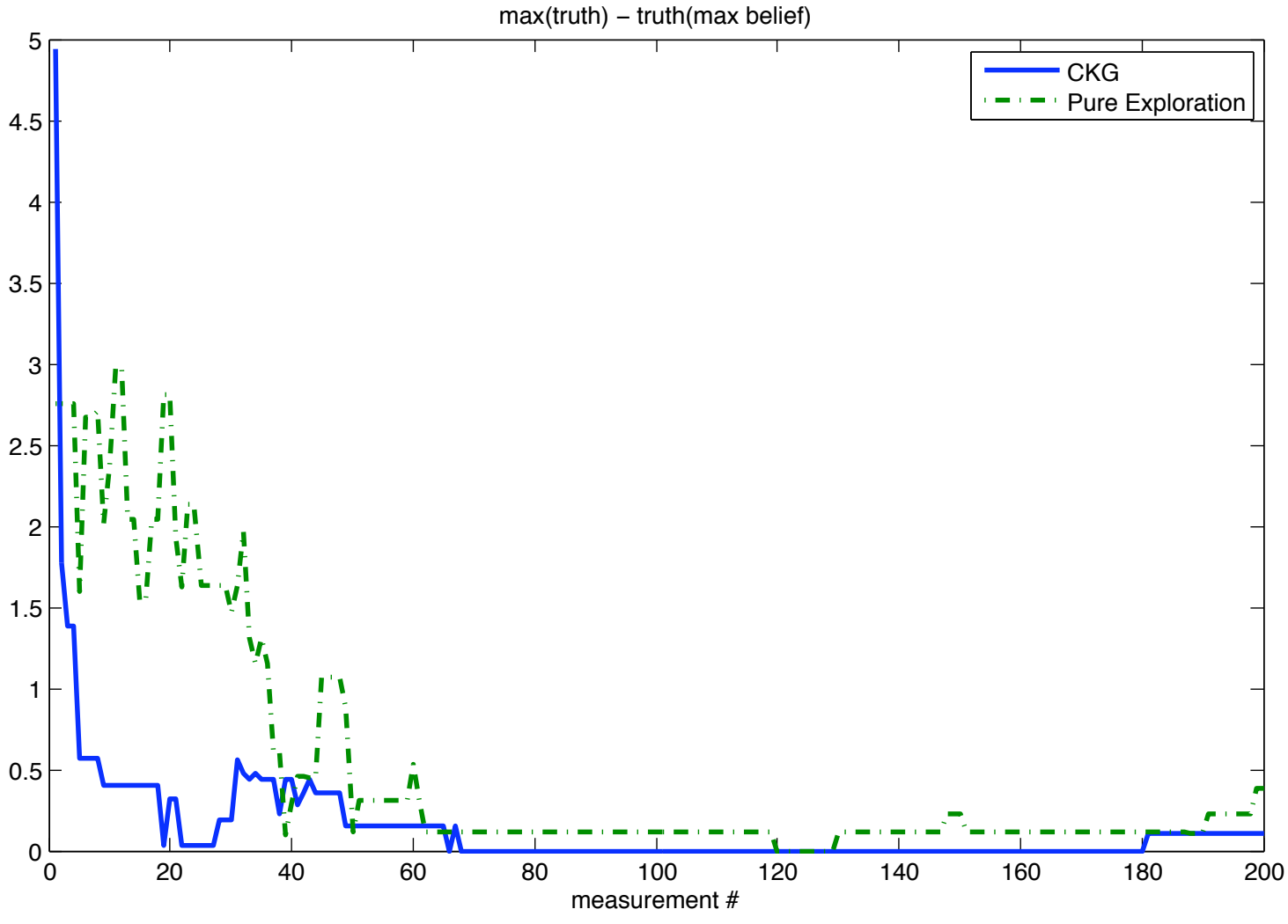
- We use Gaussian Process regression over the discrete, combinatorial, space of molecules.
  (over a discrete space, this is also called Bayesian linear regression).
- The covariance $\Sigma_0(x, x')$ of two molecules x and x' is larger when the two molecules have more substituents in common.
- This is called the Free-Wilson model in medicinal chemistry [Free and Wilson 1964].

KGCB Works Well in Tests



Average over 100 sample paths on randomly selected subsets of benzomorphan compounds of size 99.

KGCB Works Well in Tests



One sample path on the full set of 87,120 benzomorphan compounds.

Discussion: KGCB Works Well So Far...

- BGO methods work well in test problems using a chemical dataset from the literature. [Negoescu, Frazier, Powell 2011]
- Application to Ewing's sarcoma is ongoing.
- Our fingers are crossed...

Outline

Details I left out

- Choice of prior distribution.
- Monitoring the quality of the prior (model validation)
- Computational issues
- Transforming the objective function to improve model fit
- Relationship to kriging
- Other BGO methods, [Huang et al., 2006, Taddy et al., 2009, Villemonteix et al., 2009, Kleijnen et al., 2011],. . .
- Open problems. . .
- Parallelization
- Incorporating gradient information
- . . .

Software

All software is free unless otherwise noted.

- http://optimallearning.princeton.edu/ and go to "Downloadable Software"
- TOMLAB (http://tomopt.com/tomlab/) a (commercial) Matlab add-on with implementations noise-free EGO on continuous spaces.
- SPACE (http://www.schonlau.net/space.html), an implementation of EGO in C on continuous spaces.
- the matlabKG library (http://people.orie.cornell.edu/pfrazier/src.html) an implementation of the KGCB algorithm for noisy discrete problems. I am planning to improve this library, both with respect to speed and usability — if you use it, please send me an email and share your experiences.
- Software library accompanying the book by Sobester & Keane, Go to http://www.soton.ac.uk/~aijf197/ and search for "software".

More Software

- dace, a matlab kriging toolbox http://www2.imm.dtu.dk/~hbn/dace/. A Matlab library for doing kriging, which is very similar to GP regression. Assumes noise-free function evaluations, but can be easily tweaked.
- stochastic kriging, http://stochastickriging.net/. Matlab code for obtaining kriging estimates with unknown and variable sampling noise.
- For other GP regression software from the machine learning community, see http://www.gaussianprocess.org.

Introductory Reading

Brochu, E., Cora, V. M., and de Freitas, N. (2009).

A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.

Technical Report TR-2009-23, Department of Computer Science, University of British Columbia.

Forrester, A., Sobester, A., and Keane, A. (2008).

Engineering design via surrogate modelling: a practical guide.

Wiley, West Sussex, UK.

Powell, W. and Frazier, P. (2008).

Optimal Learning.

TutORials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age, pages 213–246.

Rasmussen, C. and Williams, C. (2006).

Gaussian Processes for Machine Learning.

MIT Press, Cambridge, MA.

Introductory and Advanced Reading (added after talk)

- Warren Powell and Ilya Ryzhov have a book called "Optimal Learning" that will be published in 2012.
- Some introductory (and advanced) material may be found at http://optimallearning.princeton.edu/
- (advanced reading) The KGCB algorithm for discrete and continuous spaces is introduced in [Frazier et al., 2009b, Scott et al., 2011].
- Advanced surveys and research papers may be found at http://people.orie.cornell.edu/pfrazier/

Conclusion

- BGO methods use the Bayesian posterior on the unknown function to decide where to sample next.
- They tend to require a lot of computation to decide where to sample, but reduce the overall number of samples required.
- They are very **flexible**, and the Bayesian statistical used can be tuned to new applications (non-stationary output, combinatorial feasible set, ...)

Thank You

Any questions?

Choice of $\mu_0(\cdot)$

- The Gaussian process prior is parameterized by $\mu_0(\cdot)$ and $\Sigma_0(\cdot, \cdot)$.
- How should we choose these functions?
- One common choice for $\mu_0(\cdot)$ is simply to set it to a constant β_0.
  - Typically, one estimates this constant adaptively using maximum likelihood. (Discussed later)
  - Alternatively, if one places an independent normal prior on β_0, then this can be folded back into the GP prior.
  - Coupled with typical choices for $\Sigma(\cdot, \cdot)$, this produces a prior that is stationary across the domain: for any a, the likelihood that $f(x) = a$ does not depend on x.

Choice of $\mu_0(\cdot)$

- Alternatively, if we suspect strong trends in f, we can choose a collection of basis functions $\phi_1, \ldots, \phi_K$, and set

$$\mu_0(x) = \beta_0 + \beta_1\phi_1(x) + \cdots + \beta_K\phi_m(x).$$

  - This generally does not produce a stationary prior.
  - Typically one estimates $\beta_0, \ldots, \beta_m$ using maximum likelihood.
  - Alternatively, one can place normal priors on the β_k.

Choice of $\Sigma_0(\cdot)$

We usually choose $\Sigma_0(\cdot,\cdot)$ from one of a few parametric classes of covariance functions.

- isometric Gaussian

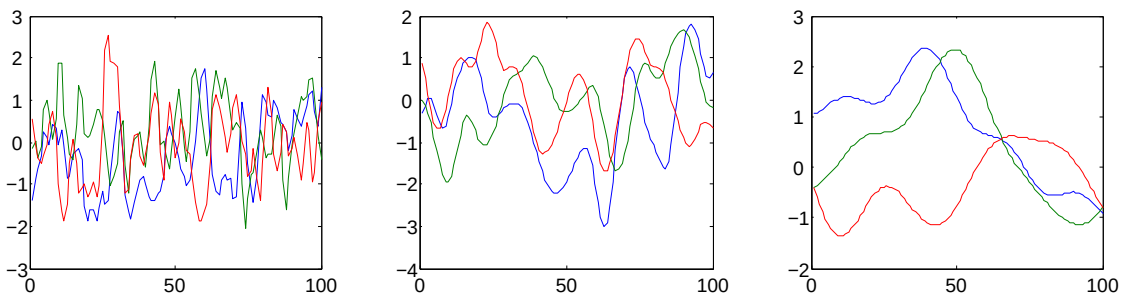
$$\Sigma_0(x,x') = \alpha_0 \exp\left(-\alpha_1 ||x-x'||_2^2\right)$$

- power exponential

$$\Sigma_0(x,x') = \alpha_0 \exp\left(-\sum_{d=1}^{D} \alpha_d |e_d \cdot (x-x')|^p\right)$$

- For others, see [Cressie, 1993, Rasmussen and Williams, 2006].

By choosing different parameter values, we can encode different beliefs in the smoothness of f.



We estimate these parameters adaptively using maximum likelihood.

Empirical Bayes Estimation of Parameters

- We have observed $x_1, \ldots, x_n$, and $y_1 = f(x_1), \ldots, y_n = f(x_n)$.
- We have a Gaussian process prior with $\mu(\cdot)$, $\Sigma(\cdot, \cdot)$.
- $\mu(\cdot)$ and $\Sigma(\cdot, \cdot)$ are parameterized in turn by a collection of parameters ν.
- To estimate ν, we calculate the density of the prior at the observed data,

  $$P(y_1, \ldots, y_n; \nu)$$

  This density is multivariate normal with a mean and covariance that depends on ν.
- We find the ν that maximizes this density, and this is our estimate.

  $$\hat{\nu} \in \arg\max_{\nu} P(y_1, \ldots, y_n; \nu)$$

- We generally update this estimate as we obtain more data.

References I

Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V., and Trosset, M. (1999).

A rigorous framework for optimization of expensive functions by surrogates.

Structural and Multidisciplinary Optimization, 17(1):1–13.

Brochu, E., Cora, V. M., and de Freitas, N. (2009).

A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.

Technical Report TR-2009-23, Department of Computer Science, University of British Columbia.

Calvin, J. and Zilinskas, A. (2002).

One-dimensional Global Optimization Based on Statistical Models.

Nonconvex Optimization and its Applications, 59:49–64.

Calvin, J. and Zilinskas, A. (2005).

One-Dimensional global optimization for observations with noise.

Computers & Mathematics with Applications, 50(1-2):157–169.

Cressie, N. (1993).

Statistics for Spatial Data, revised edition.

Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley Interscience, New York.

Forrester, A., Keane, A., and Bressloff, N. (2006).

Design and Analysis of" Noisy" Computer Experiments.

AIAA Journal, 44(10):2331–2339.

Forrester, A., Sobester, A., and Keane, A. (2008).

Engineering design via surrogate modelling: a practical guide.

Wiley, West Sussex, UK.

References II

Frazier, P., Powell, W., and Simão, H. (2009a).

Simulation model calibration with correlated knowledge-gradients.
In *Winter Simulation Conference Proceedings, 2009*. Winter Simulation Conference.

Frazier, P., Powell, W. B., and Dayanik, S. (2009b).

The knowledge gradient policy for correlated normal beliefs.
INFORMS Journal on Computing, 21(4):599–613.

Huang, D., Allen, T., Notz, W., and Miller, R. (2006).

Sequential kriging optimization using multiple-fidelity evaluations.
Structural and Multidisciplinary Optimization, 32(5):369–382.

Jones, D., Schonlau, M., and Welch, W. (1998).

Efficient Global Optimization of Expensive Black-Box Functions.
Journal of Global Optimization, 13(4):455–492.

Kleijnen, J., van Beers, W., and van Nieuwenhuyse I. (2011).

Expected improvement in efficient global optimization through bootstrapped kriging.
Journal of Global Optimization, pages 1–15.

Kushner, H. J. (1964).

A new method of locating the maximum of an arbitrary multi- peak curve in the presence of noise.
Journal of Basic Engineering, 86:97–106.

Mockus, J. (1972).

On bayesian methods for seeking the extremum.
Automatics and Computers (Avtomatika i Vychislitelnayya Tekchnika), 4(1):53–62.
(in Russian).

References III

Mockus, J. (1989).

Bayesian approach to global optimization: theory and applications.
Kluwer Academic, Dordrecht.

Mockus, J., Tiesis, V., and Zilinskas, A. (1978).

The application of Bayesian methods for seeking the extremum.
In Dixon, L. and Szego, G., editors, *Towards Global Optimisation*, volume 2, pages 117–129. Elsevier Science Ltd., North Holland, Amsterdam.

Negoescu, D., Frazier, P., and Powell, W. (2011).

The knowledge gradient algorithm for sequencing experiments in drug discovery.
INFORMS Journal on Computing, 23(1).

Powell, W. and Frazier, P. (2008).

Optimal Learning.
TutORials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age, pages 213–246.

Rasmussen, C. and Williams, C. (2006).

Gaussian Processes for Machine Learning.
MIT Press, Cambridge, MA.

Regis, R. and Shoemaker, C. (2005).

Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions.
Journal of Global Optimization, 31(1):153–171.

Scott, W., Frazier, P. I., and Powell, W. B. (2011).

The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression.
SIAM Journal on Optimization, 21:996–1026.

References IV

Stuckman, B. (1988).

A global search method for optimizing nonlinear systems.

Systems, Man and Cybernetics, IEEE Transactions on, 18(6):965–977.

Taddy, M., Lee, H., Gray, G., and Griffin, J. (2009).

Bayesian guided pattern search for robust local optimization.

Technometrics, 51(4):389–401.

Villemonteix, J., Vazquez, E., and Walter, E. (2009).

An informational approach to the global optimization of expensive-to-evaluate functions.

Journal of Global Optimization, 44(4):509–534.

Yang, W., Feinstein, J., and Marsden, A. (2010).

Constrained optimization of an idealized y-shaped baffle for the fontan surgery at rest and exercise.

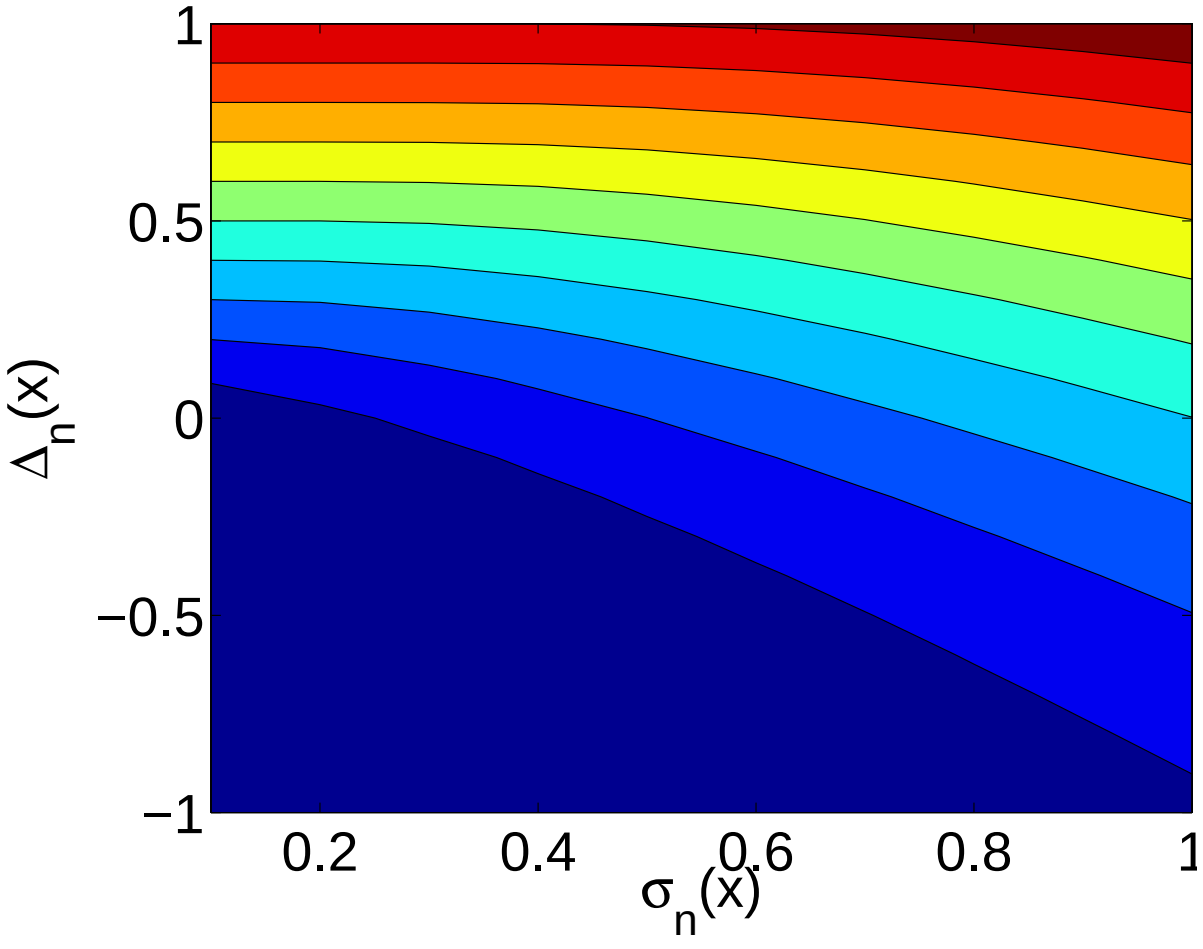
Computer methods in applied mechanics and engineering, 199(33-36):2135–2149.

EI Trades Exploration vs. Exploitation

- $\mathrm{EI}_n(x) = [\Delta_n(x)]^+ + \sigma_n(x)\varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(-\frac{|\Delta_n(x)|}{\sigma_n(x)}\right)$
- $\mathrm{EI}_n(x)$ is determined by $\Delta_n(x) = \mu_n(x) - f_n^*$ and $\sigma_n(x)$.
- $\mathrm{EI}_n(x)$ increases as $\Delta_n(x)$ increases.
  - Measure where $f(x)$ seems large. (**Exploitation**)
- $\mathrm{EI}_n(x)$ increases as $\sigma_n(x)$ increases.
  - Measure where we are uncertain about $f(x)$. (**Exploration**)

EI Trades Exploration vs. Exploitation

- $\mathrm{EI}_n(x)$ is bigger when $\mu_n(x)$ is bigger.
- $\mathrm{EI}_n(x)$ is bigger when $\sigma_n(x)$ is bigger.
- Below is a contour plot of $\mathrm{EI}_n(x)$. Red is bigger EI.



Knowledge-Gradient with Correlated Beliefs (KGCB)

- Call this modified expected improvement the knowledge-gradient (KG) factor

$$\mathrm{KG}_n(x) = \mathbb{E}_n\left[\mu^*_{n+1} - \mu^*_n \mid x_{n+1} = x\right].$$

- The KGCB policy measures at the point with the largest KG factor.

$$x_{n+1} \in \arg\max_x \mathrm{KG}_n(x).$$