Bayesian Optimization. Active Learning. Adaptive Design of Experiments

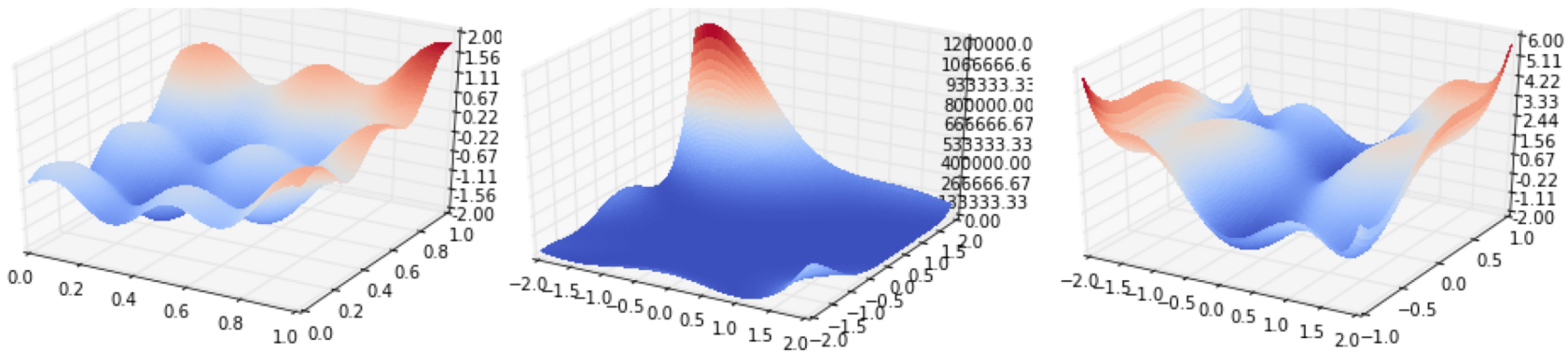
Evgeny Burnaev

Skoltech, Moscow, Russia

Outline

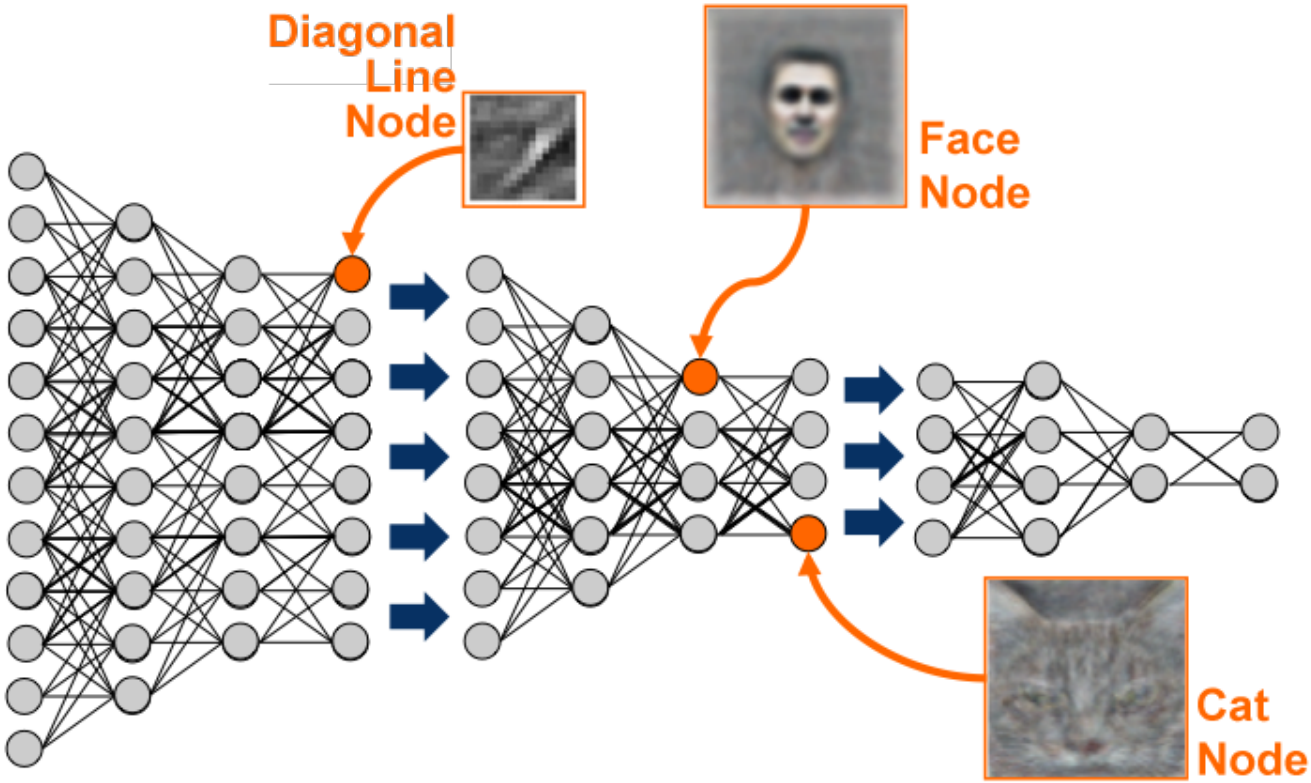
Consider a "well behaved" function $f : \mathcal{X} \to \mathbb{R}$, with $\mathcal{X} \subseteq \mathbb{R}^d$ being a compact set

$$\mathbf{x}_{\min} = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$



- f is explicitly unknown and multimodal
- Evaluations of f may be perturbed
- Evaluations of f are expensive $\Rightarrow$
  - Gradient and Hessian are not computable
  - Grid search is not possible

Parameter tuning in ML algorithms



- Number of layers/units per layer
- Types of each layer
- Regularization coefficients
- Learning rates, etc.

Parameter tuning in ML algorithms: Example of DNN

Input $\mathbf{x}$:

- Number of layers/units per layer
- Types of each layer
- Regularization coefficients
- Learning rates, etc.

Output: $f(\mathbf{x})$

- Deep Neural Network accuracy
- Estimated using cross-validation and/or test set
- Very time-consuming

Other expensive functions

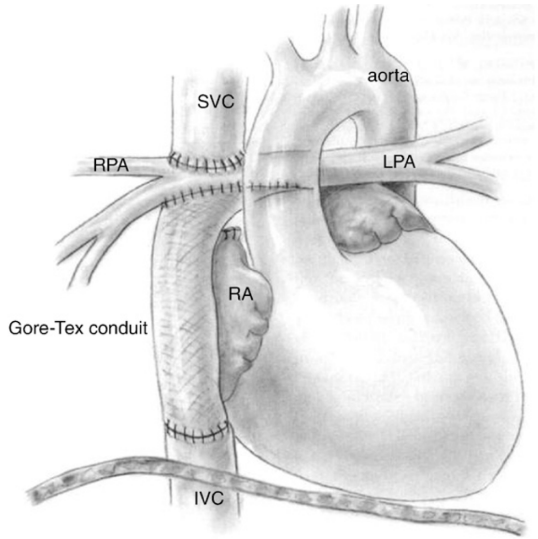


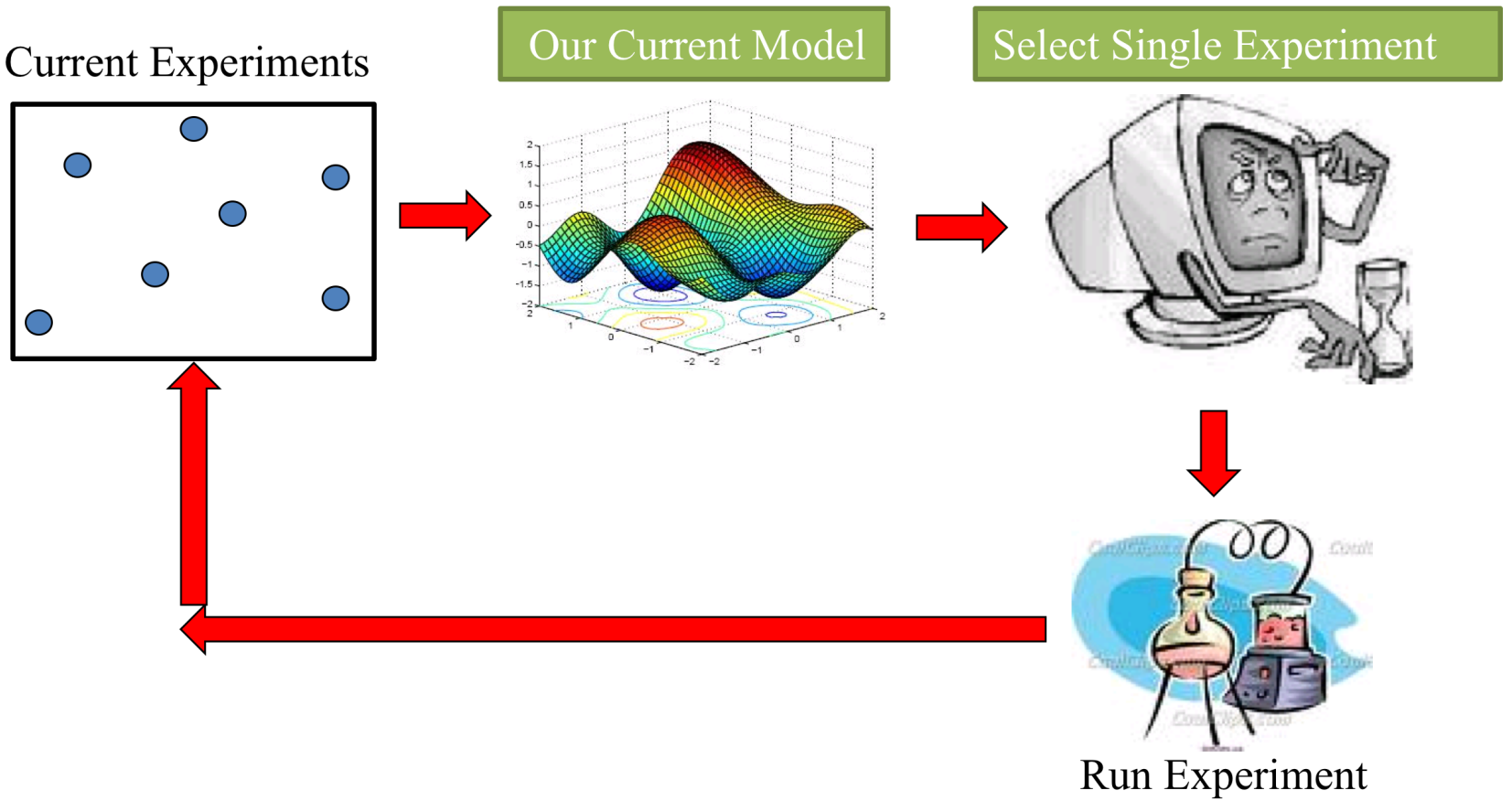
Fig. 1. Extracardiac total cavopulmonary connection. The IVC is disconnected from the right atrium (RA) and connected to the PAs via a Gore-Tex conduit. Figure taken from Reddy et al. [13].

- Design of grafts to be used in heart surgery
- Design of aeorodynamic structures, e.g., cars, airplanes
- Calibrating parameters of complex physical models to experimental data
- A/B testing data to optimize the web design to maximize sign-ups, downloads, purchases, etc.

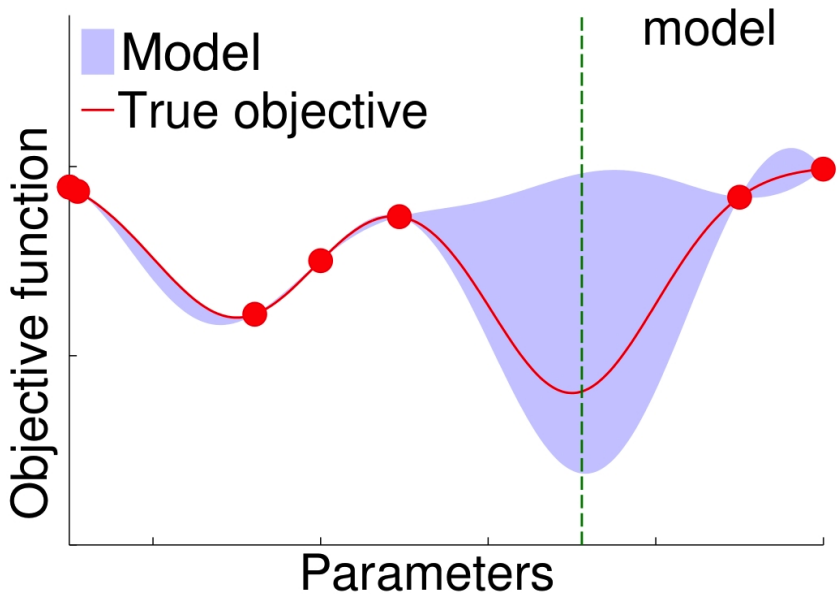
NB! There exists commercial services for optimizing black-box functions: SIGOPT, Google Vizier, etc.

Big Picture

- Since Running experiment is very expensive we use BO
- Select one experiment to run at a time based on results of previous experiments

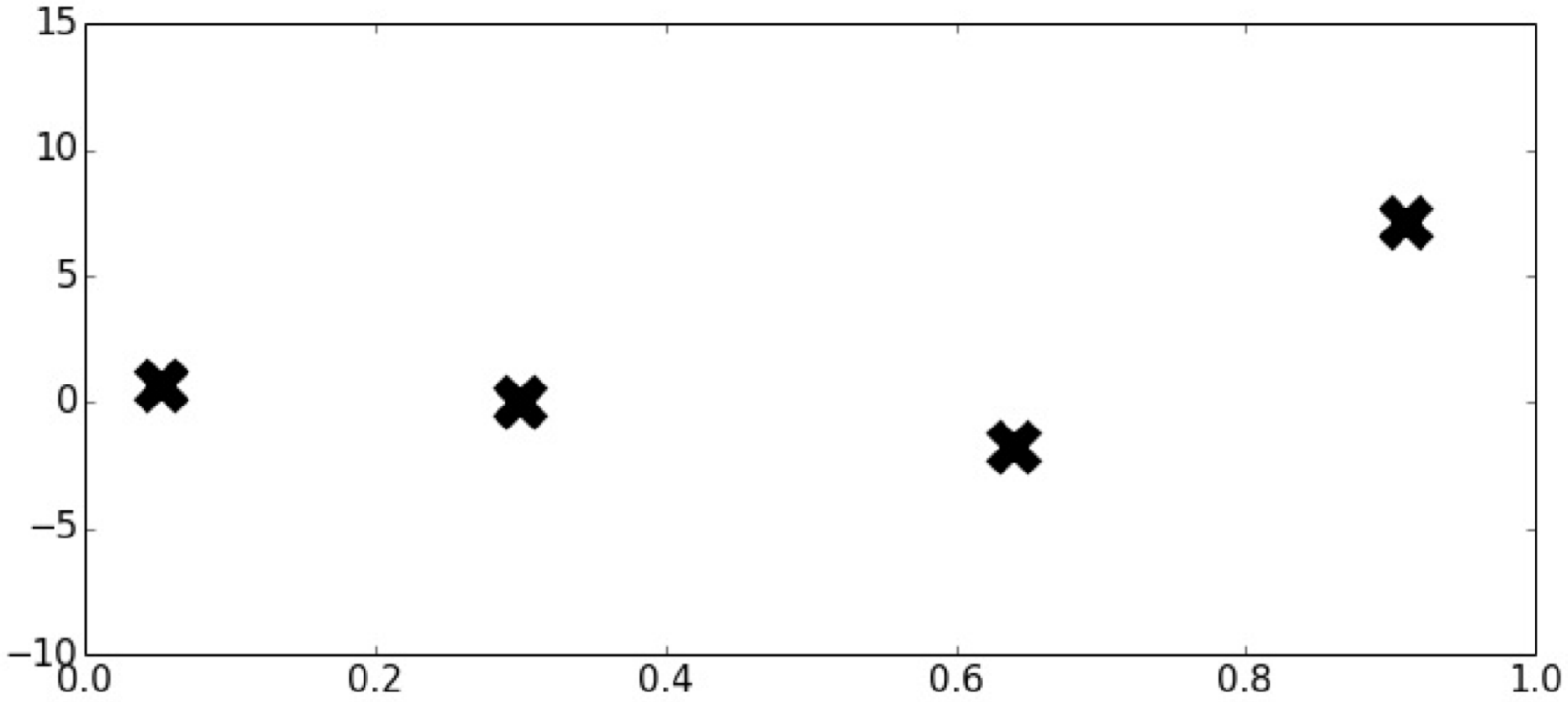


- In high-dimensional case we need many functions evaluations
- Often each evaluation is costly, e.g. in case of experiments



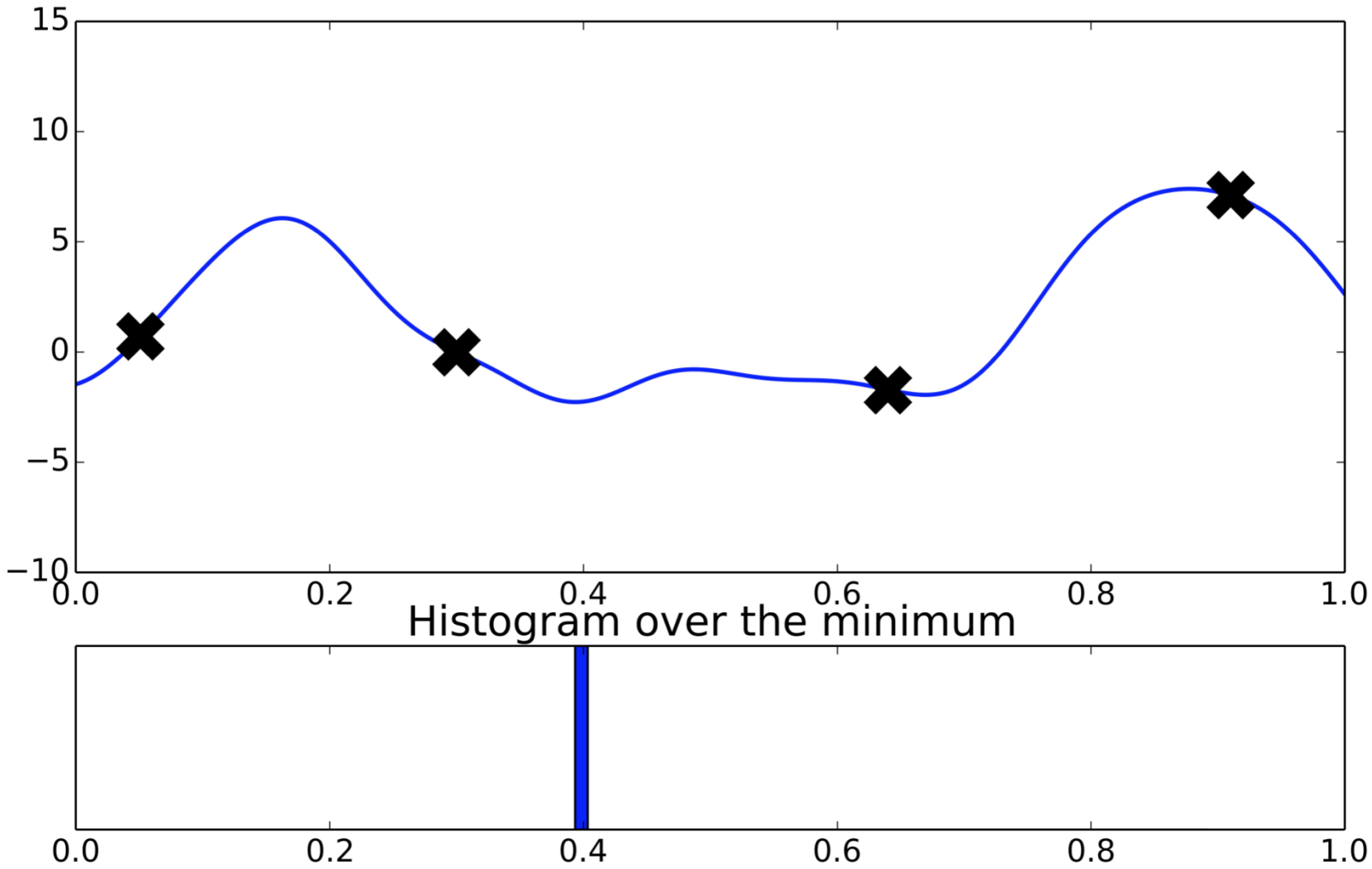
- Error bars are needed to see if a region is still promising

Typical situation

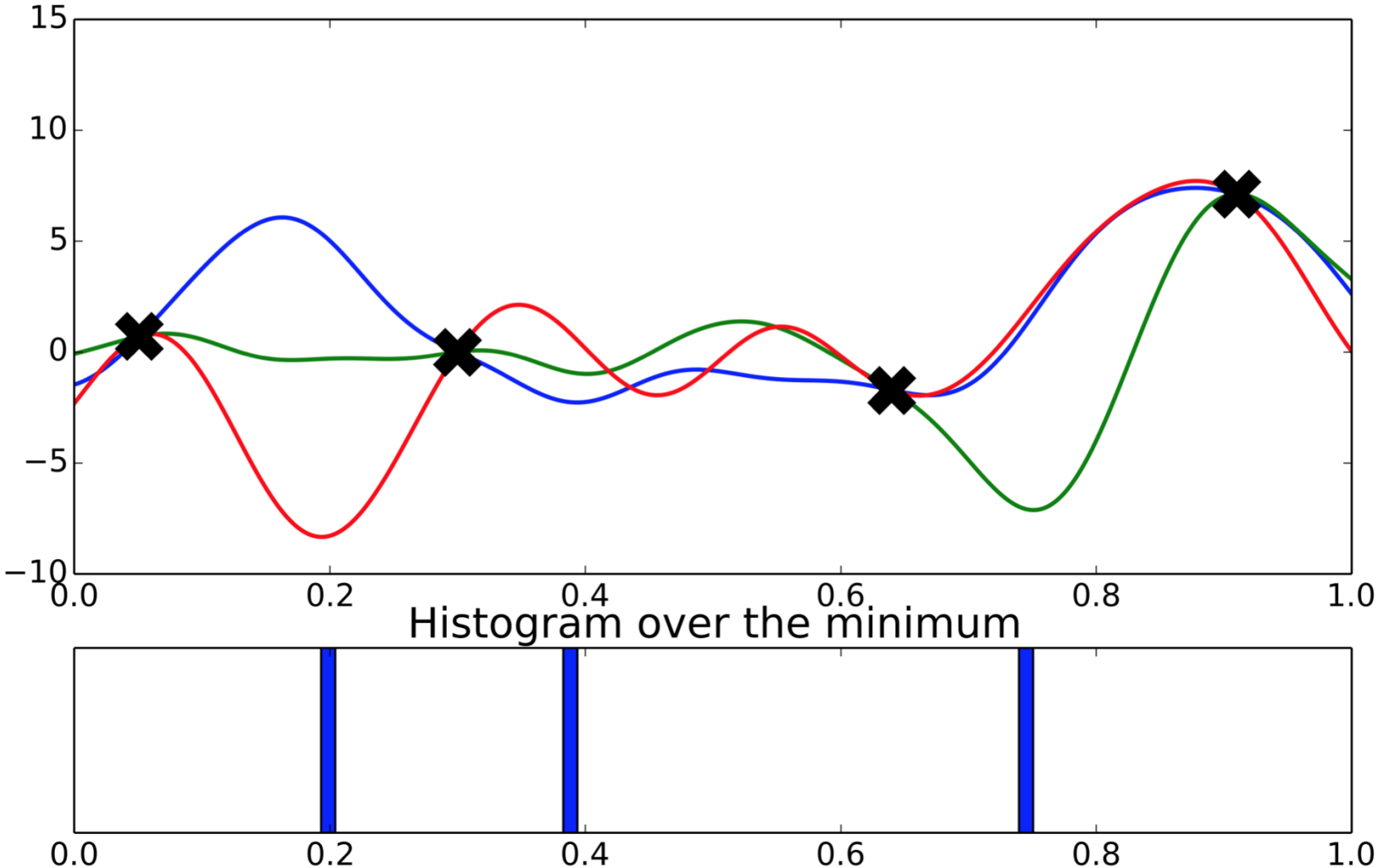


Where is the minimum of f?
Where should we evaluate the function next?

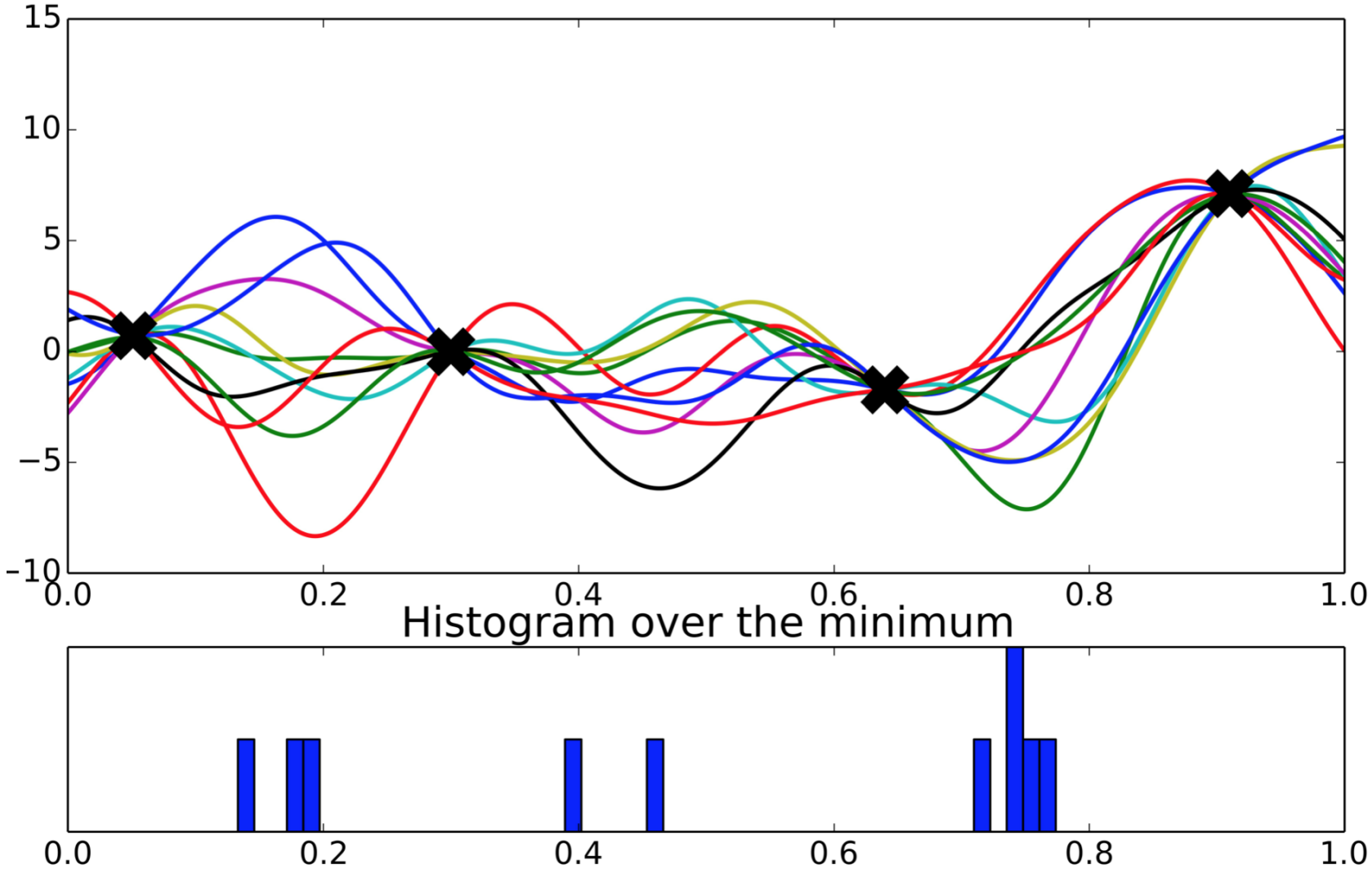
Intuition: one curve



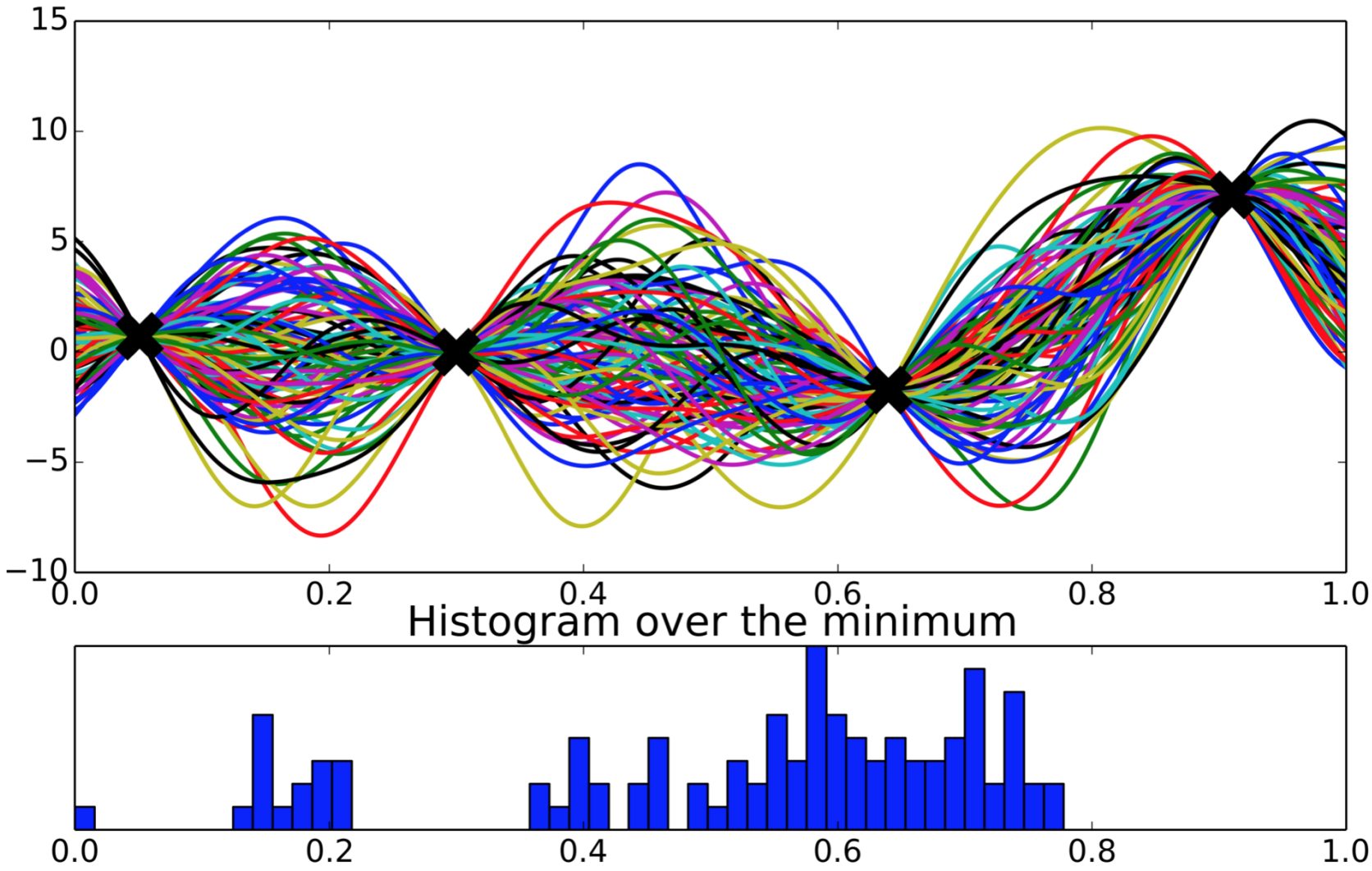
Intuition: three curves



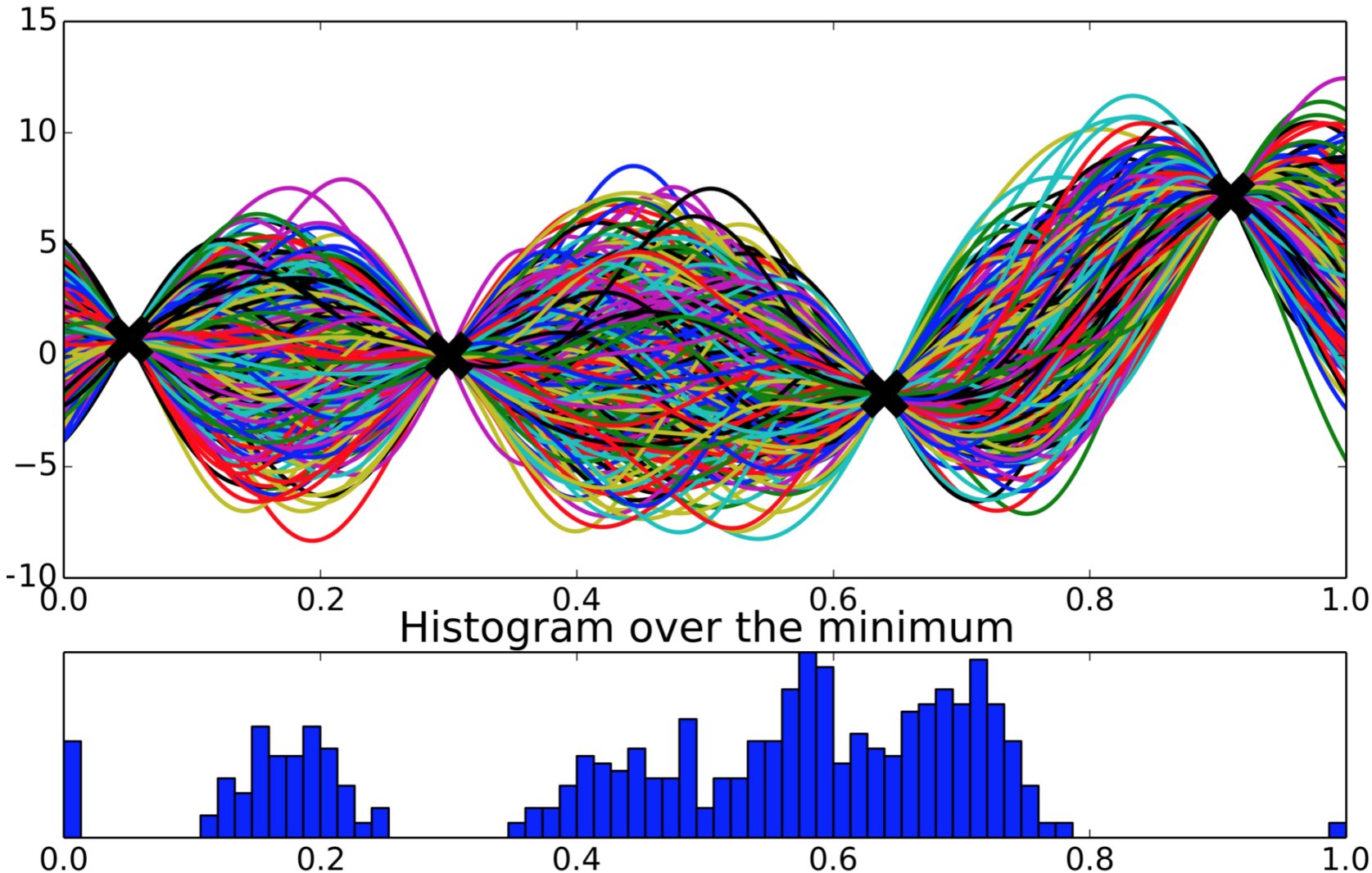
Intuition: ten curves



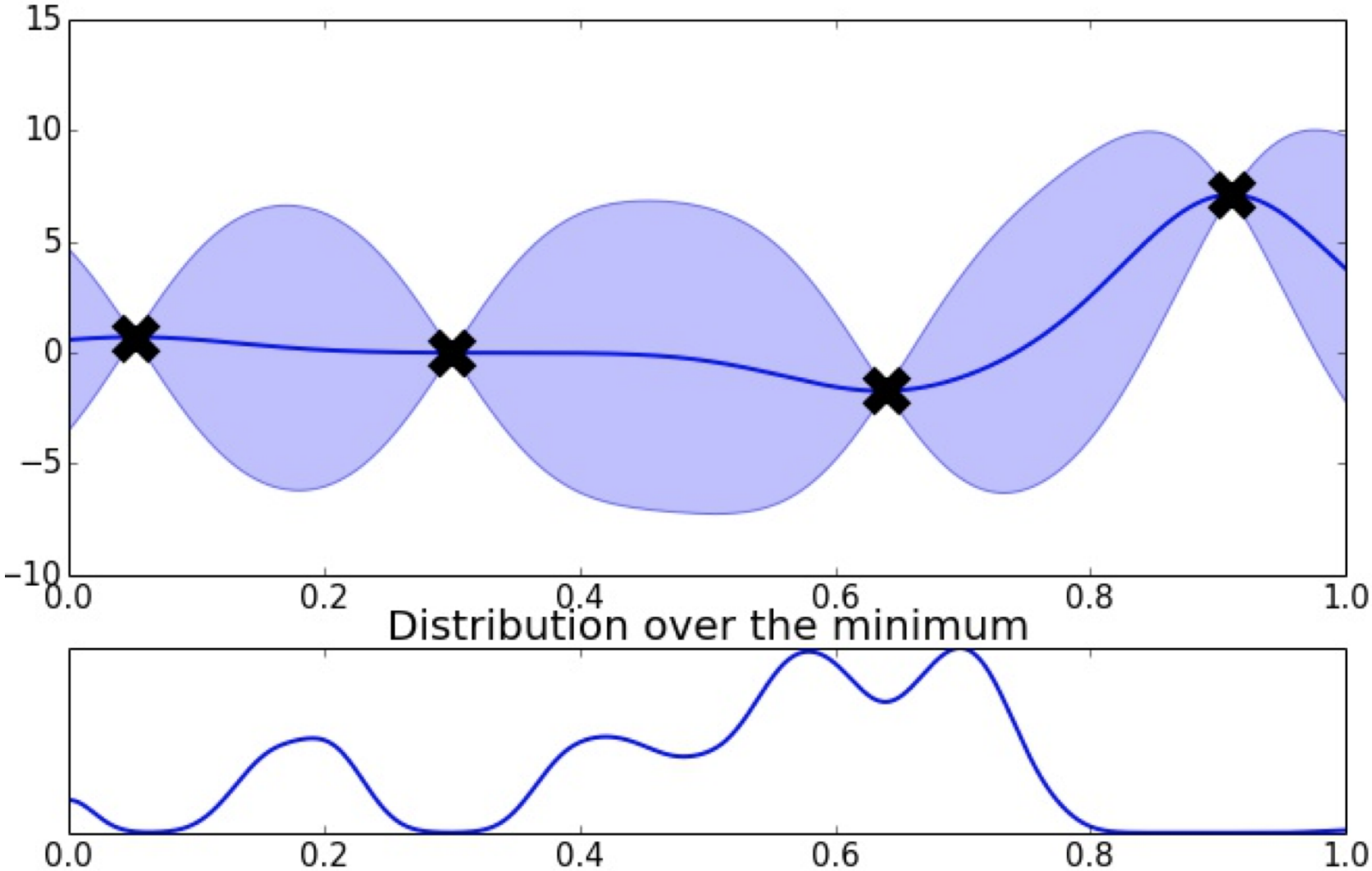
Intuition: hundred curves



Intuition: many curves



Intuition: infinite number of curves

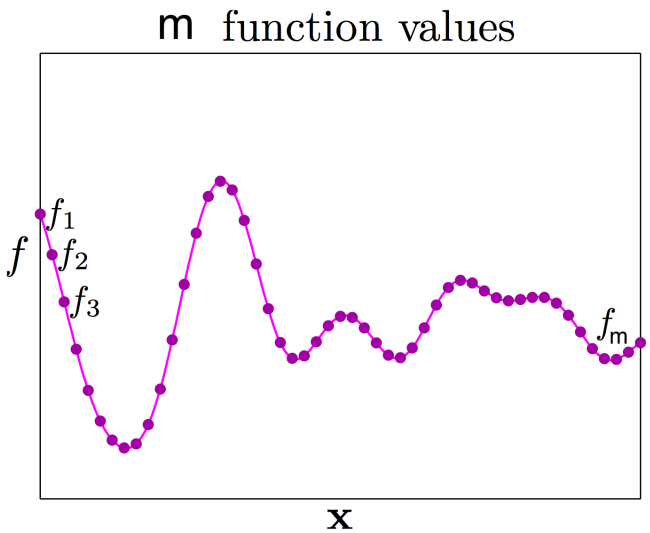


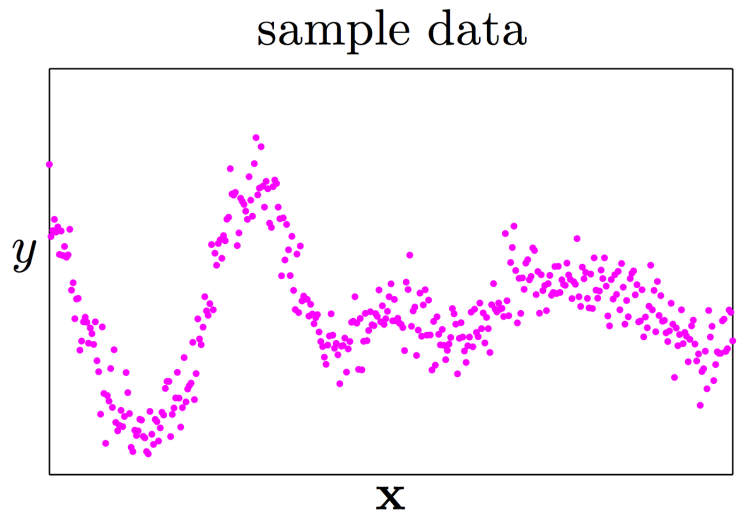
Formalization

- We made prior assumption about f
- Information about the minimum is now encoded in a new function (the probability distribution $p_{\min}$ over the minimum in this case)
- We can use $p_{\min}$ (or a functional of it) to decide where to sample next
- Other functions to encode relevant information about the minimum are possible, e. g. the "marginal expected gain" at each location.

- Training data set $S_m = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$
- Model:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i,$$
$$f \sim \mathcal{GP}(\cdot|0, K), \text{ with } K(\mathbf{x}, \mathbf{x}') = \operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}')),$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ is a white noise}$$





- Training data set $S_m = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$
- Model:

$$
\begin{aligned}
y_i &= f(\mathbf{x}_i) + \varepsilon_i \\
f &\sim \mathcal{GP}(\cdot|0, K) \\
\varepsilon_i &\sim \mathcal{N}(0, \sigma^2) \text{ is a white noise}
\end{aligned}
$$

- The prior is

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

- The noise model, or likelihood is

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \sigma^2 \mathrm{I}_m\right)$$

- Integrating over the function variables $\mathbf{f}$ we get the marginal likelihood

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}) d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0},\, \mathbf{K} + \sigma^2 \mathrm{I}_m)$$

- Let us denote input test point as $\mathbf{x}_*$, and output

$$y_* = f_* + \varepsilon_*, \;\; f_* = f(\mathbf{x}_*)$$

- Consider joint training and test marginal likelihood

$$p(\mathbf{y}, f_*) = \mathcal{N}\left(\begin{array}{c} \mathbf{y} \\ f(\mathbf{x}_*) \end{array} \middle| \mathbf{0}, \left[\begin{array}{cc} \mathbf{K} + \sigma^2 \mathrm{I}_m & \mathbf{k}_* \\ \mathbf{k}_*^\top & K_{**} \end{array} \right] \right),$$

  where $\mathbf{k}_* = \{K(\mathbf{x}_*, \mathbf{x}_i)\}_{i=1}^m$ and $K_{**} = K(\mathbf{x}_*, \mathbf{x}_*)$
- What we know about noiseless value $f(\mathbf{x}_*)$?

- Joint training and test marginal likelihood

$$p(\mathbf{y}, f_*) = \mathcal{N}\left(\begin{array}{c} \mathbf{y} \\ f(\mathbf{x}_*) \end{array} \middle| \mathbf{0}, \left[\begin{array}{cc} \mathbf{K} + \sigma^2 \mathrm{I}_m & \mathbf{k}_* \\ \mathbf{k}_*^\top & K_{**} \end{array} \right] \right),$$

  where $\mathbf{k}_* = \{K(\mathbf{x}_*, \mathbf{x}_i)\}_{i=1}^m$ and $K_{**} = K(\mathbf{x}_*, \mathbf{x}_*)$

- Condition on training outputs $\mathbf{y}$ we get

$$p(f_*|\mathbf{y}) = \mathcal{N}\left(f_*|\mu_*, \sigma_*^2\right),$$

  where

$$\begin{aligned} \mu_*(\mathbf{x}_*) &= \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y} = \\ &= \sum_{i=1}^m \alpha_i K(\mathbf{x}_*, \mathbf{x}_i) \text{ with } \boldsymbol{\alpha} = [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y} \quad \text{(aka KRR)} \\ \sigma_*^2(\mathbf{x}_*) &= K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_* \end{aligned}$$

- $p(y_*|\mathbf{y}) = \mathcal{N}\left(y_*|\mu_*, \sigma_*^2 + \sigma^2\right)$ predicts what we'll see next

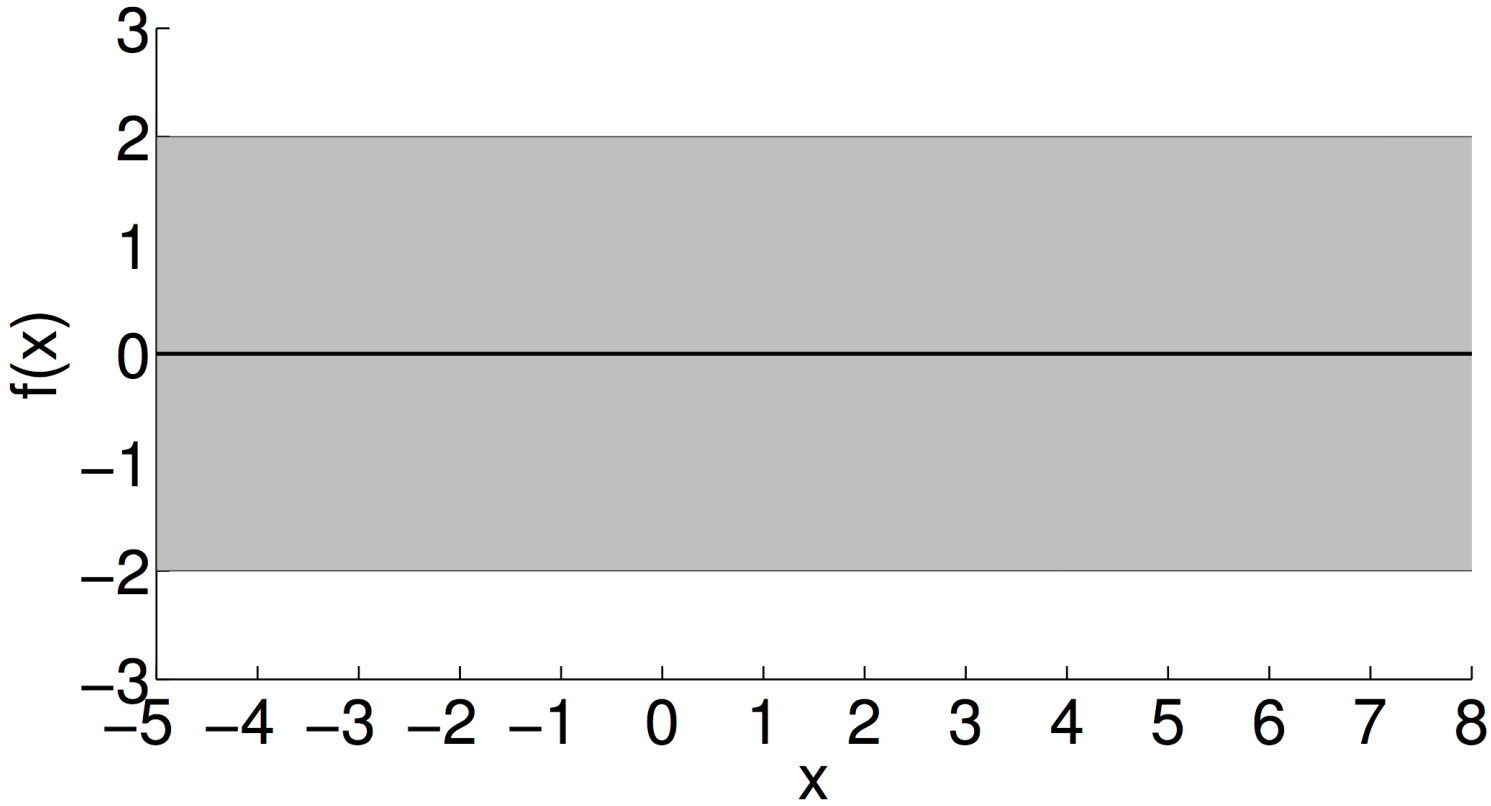


Figure – **Prior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \varnothing] = \mu_*(\mathbf{x}_*) = 0$$
$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \varnothing] = \sigma_*^2(\mathbf{x}_*) = K(\mathbf{x}_*, \mathbf{x}_*)$$

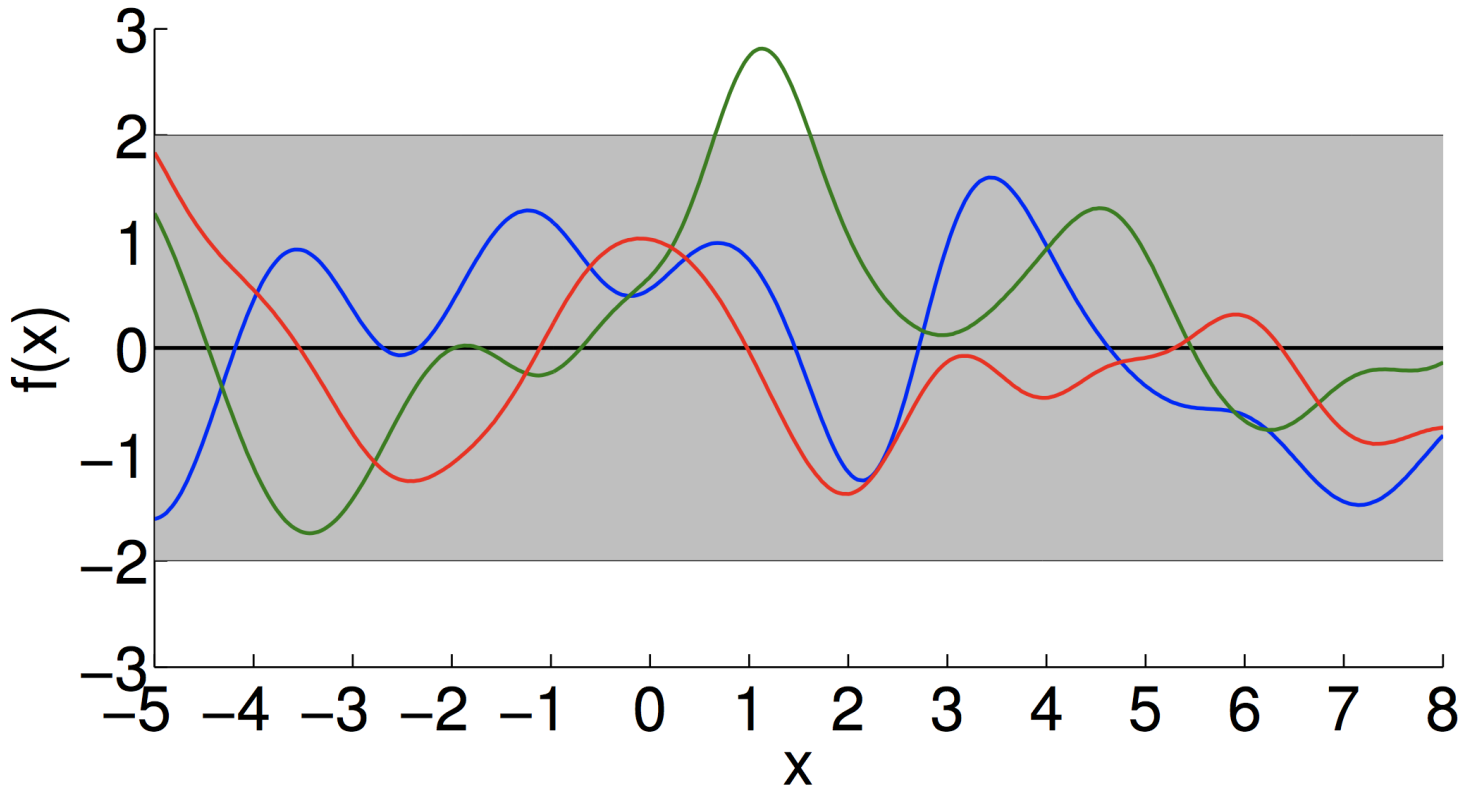


Figure – **Prior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \varnothing] = \mu_*(\mathbf{x}_*) = 0$$
$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \varnothing] = \sigma_*^2(\mathbf{x}_*) = K(\mathbf{x}_*, \mathbf{x}_*)$$

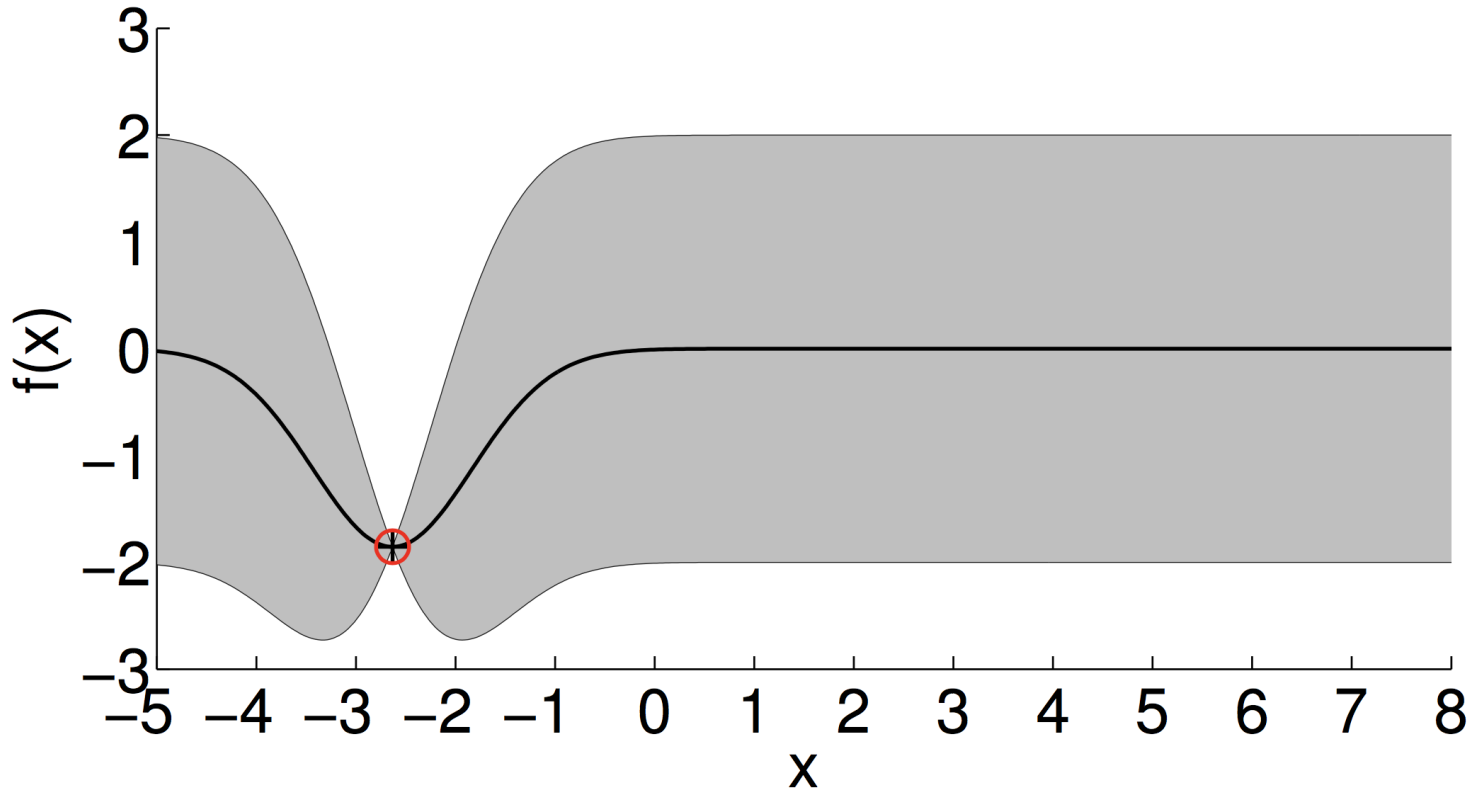


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

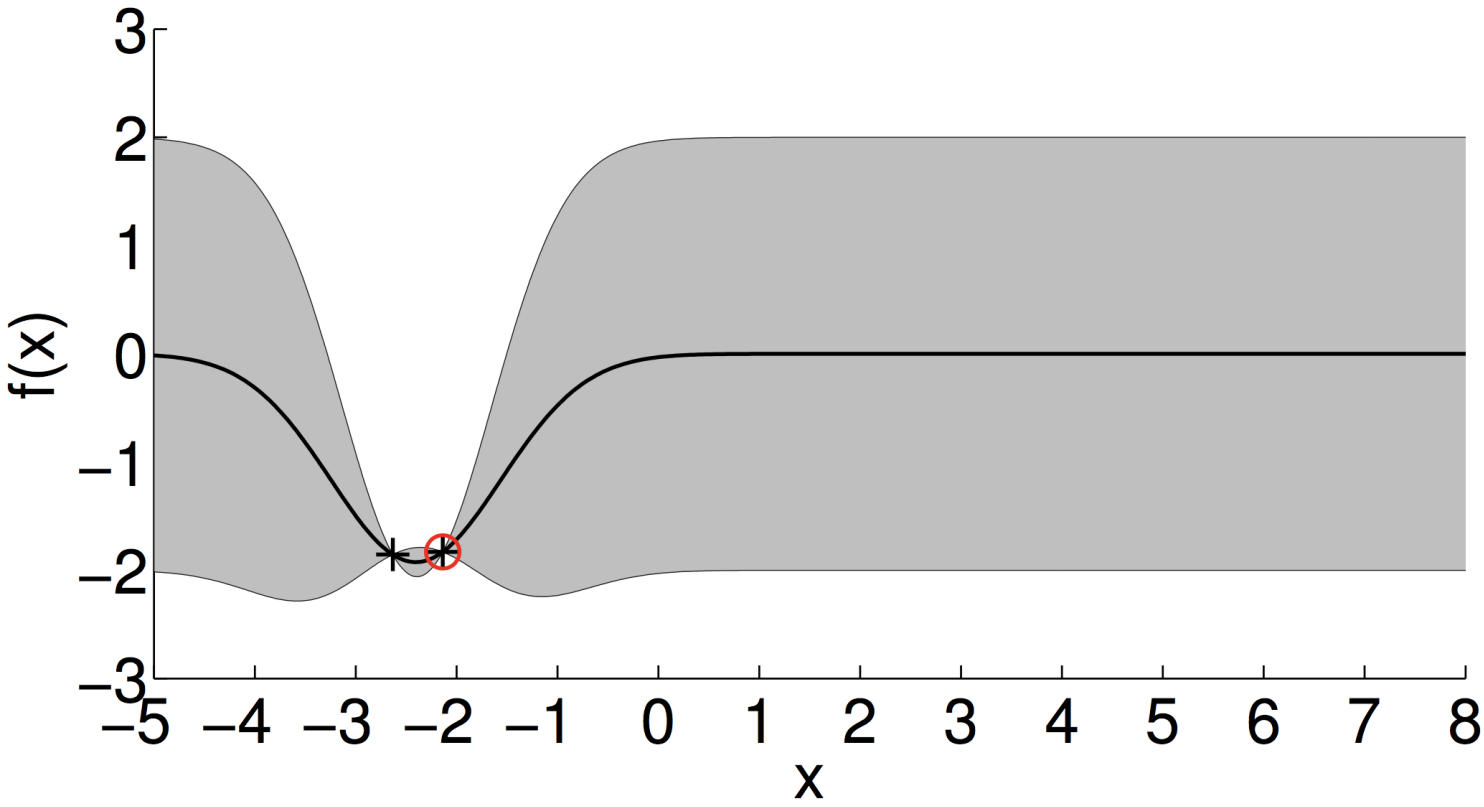


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

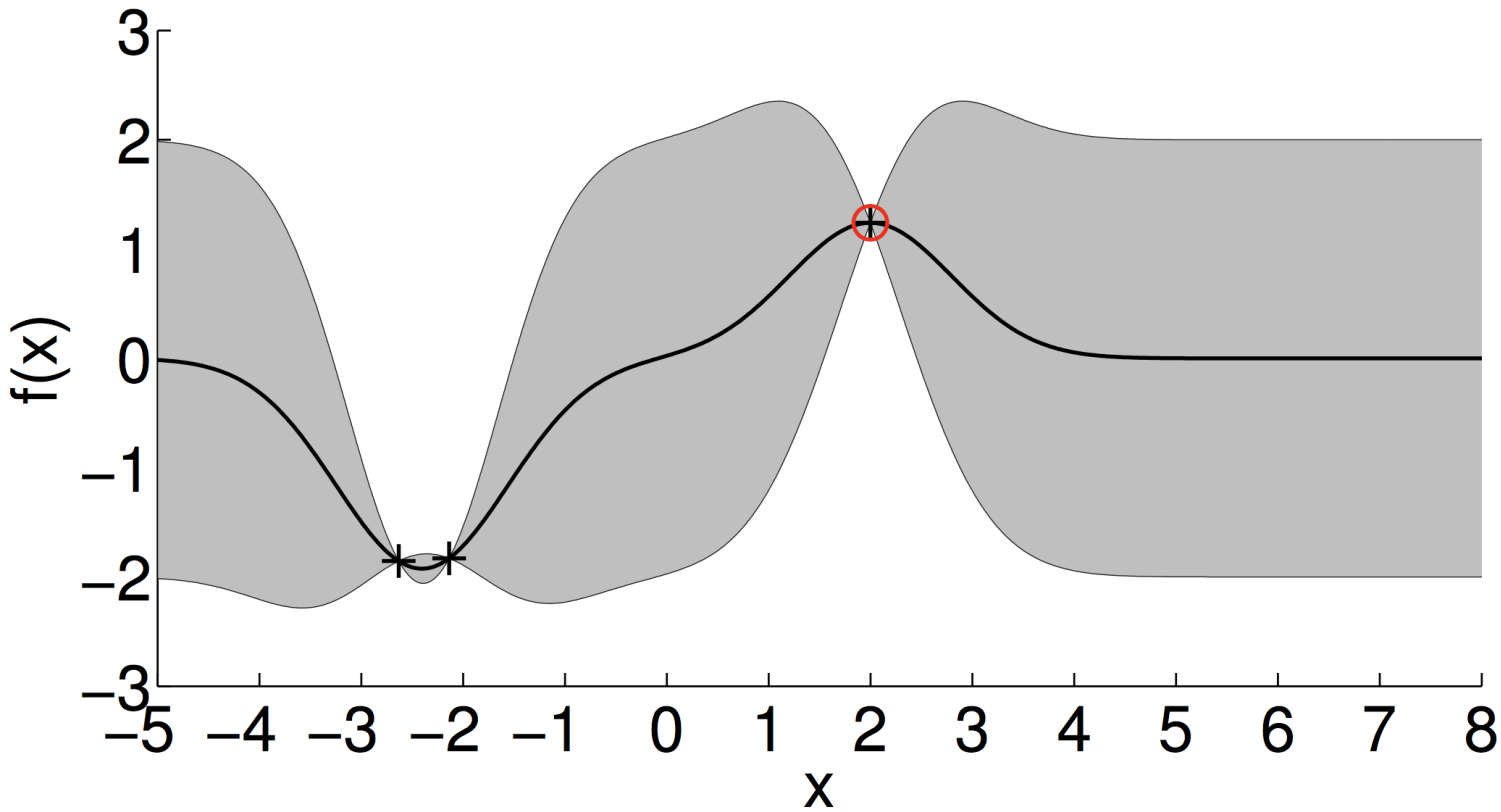


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

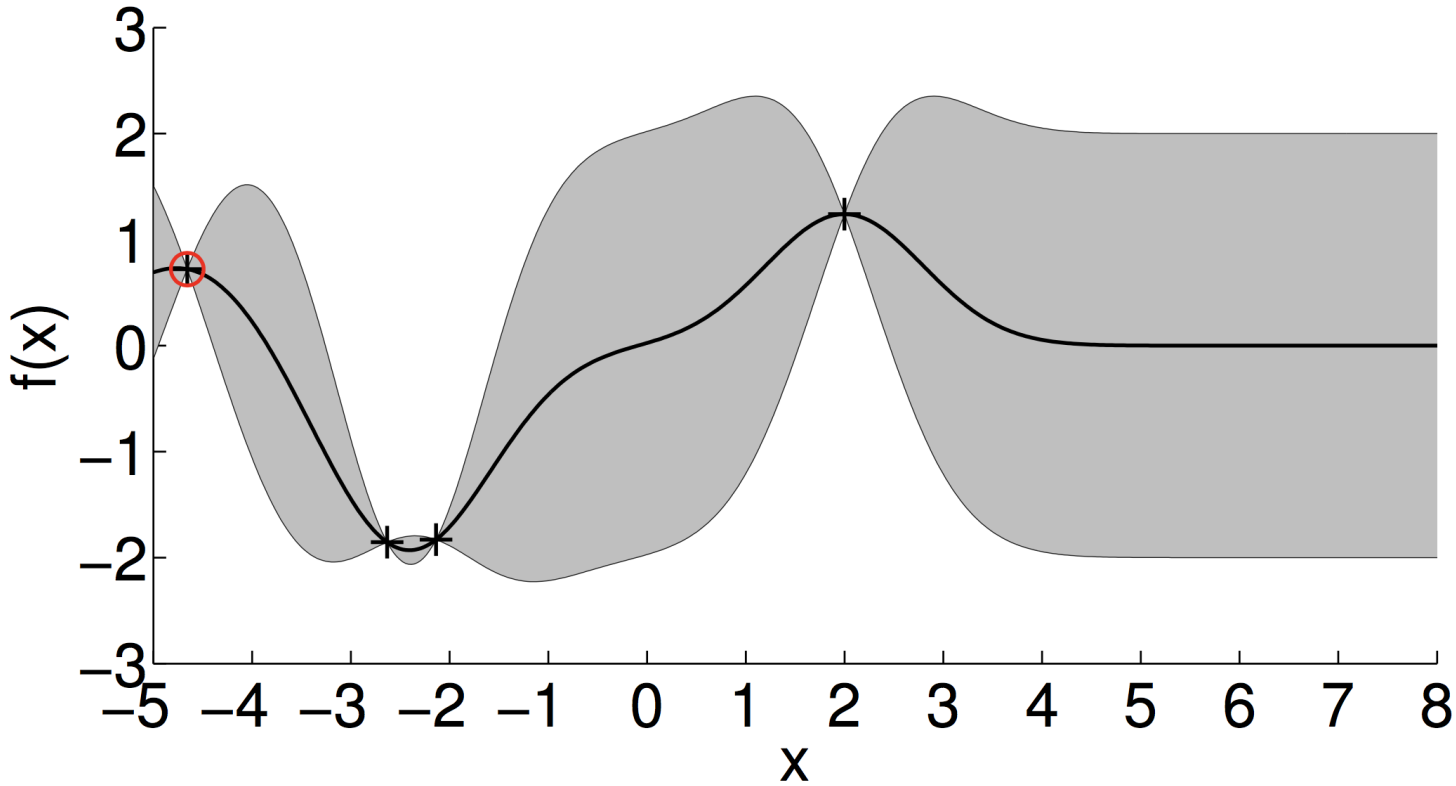


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

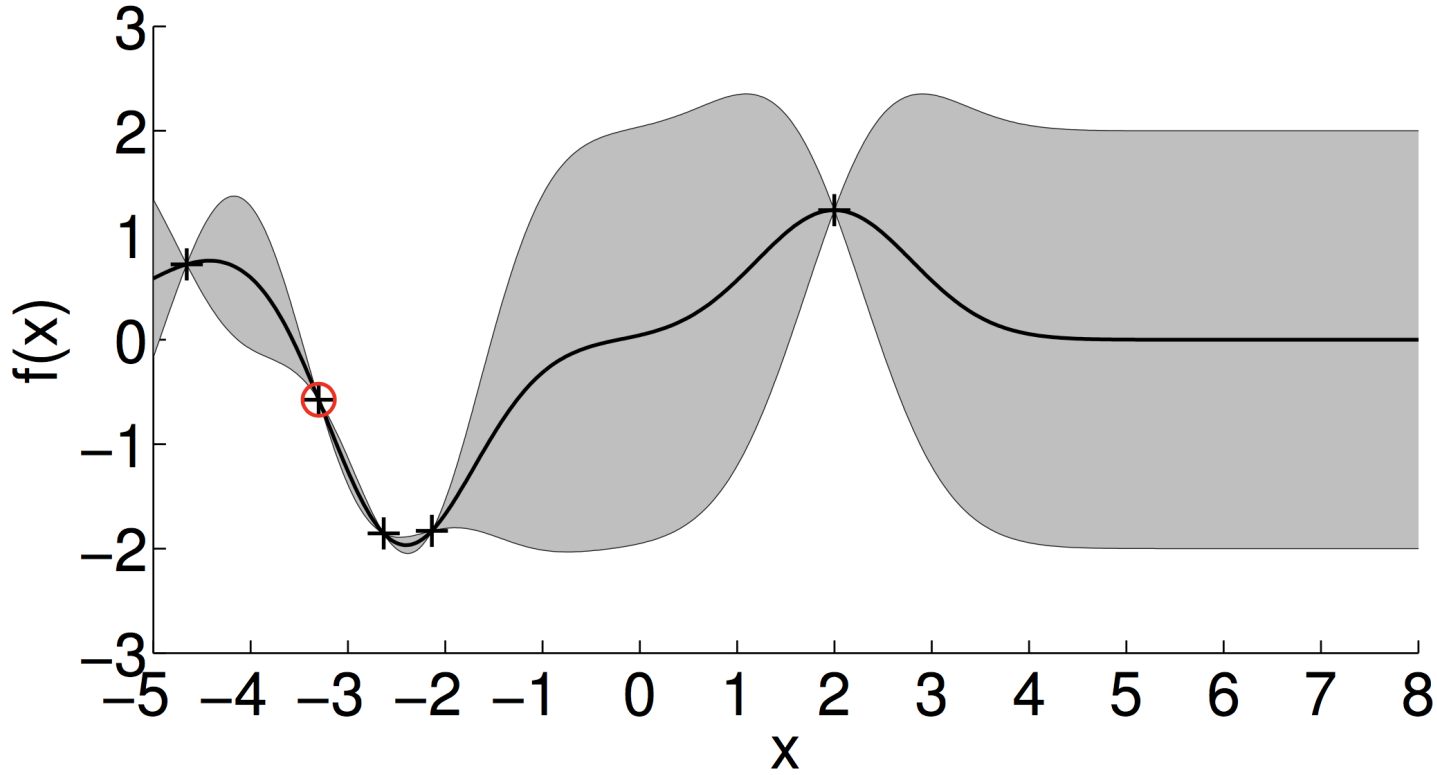


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

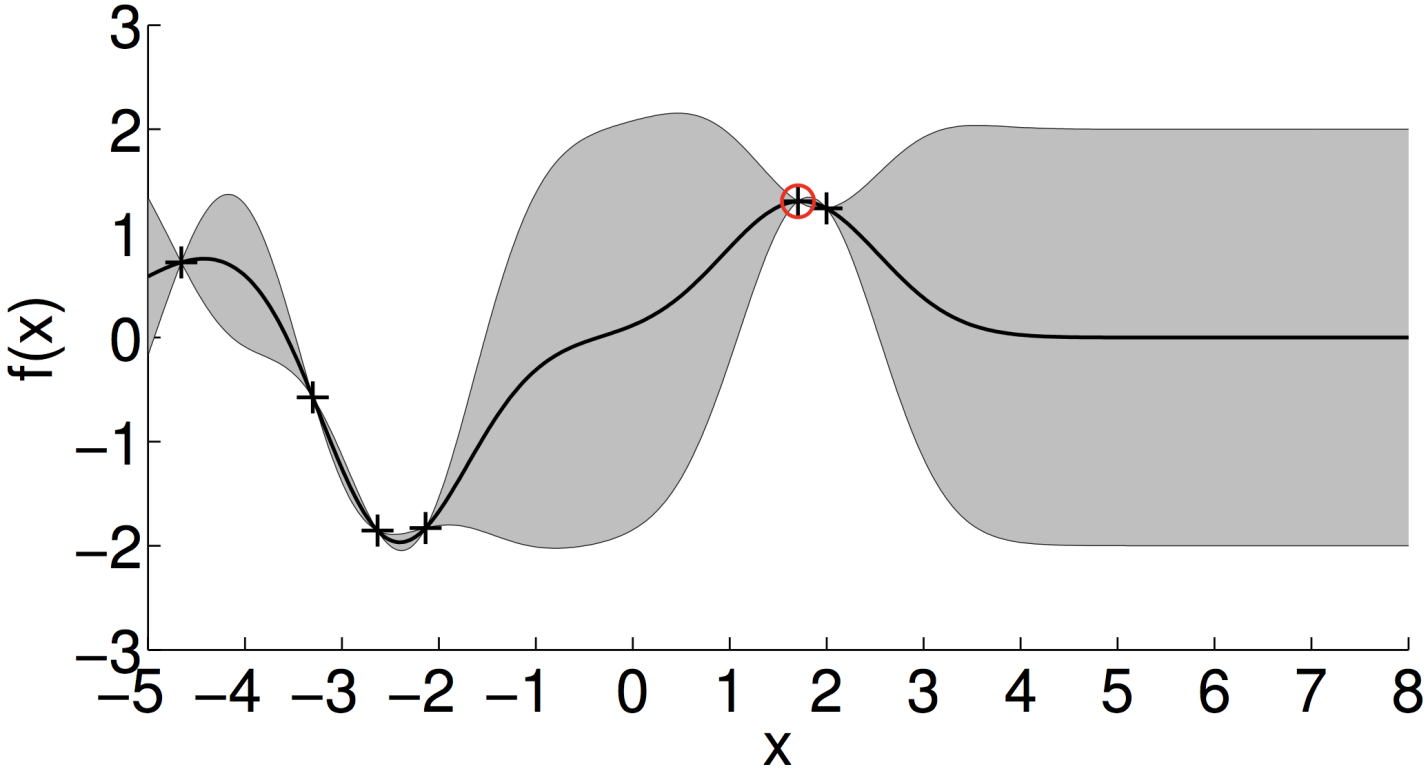


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

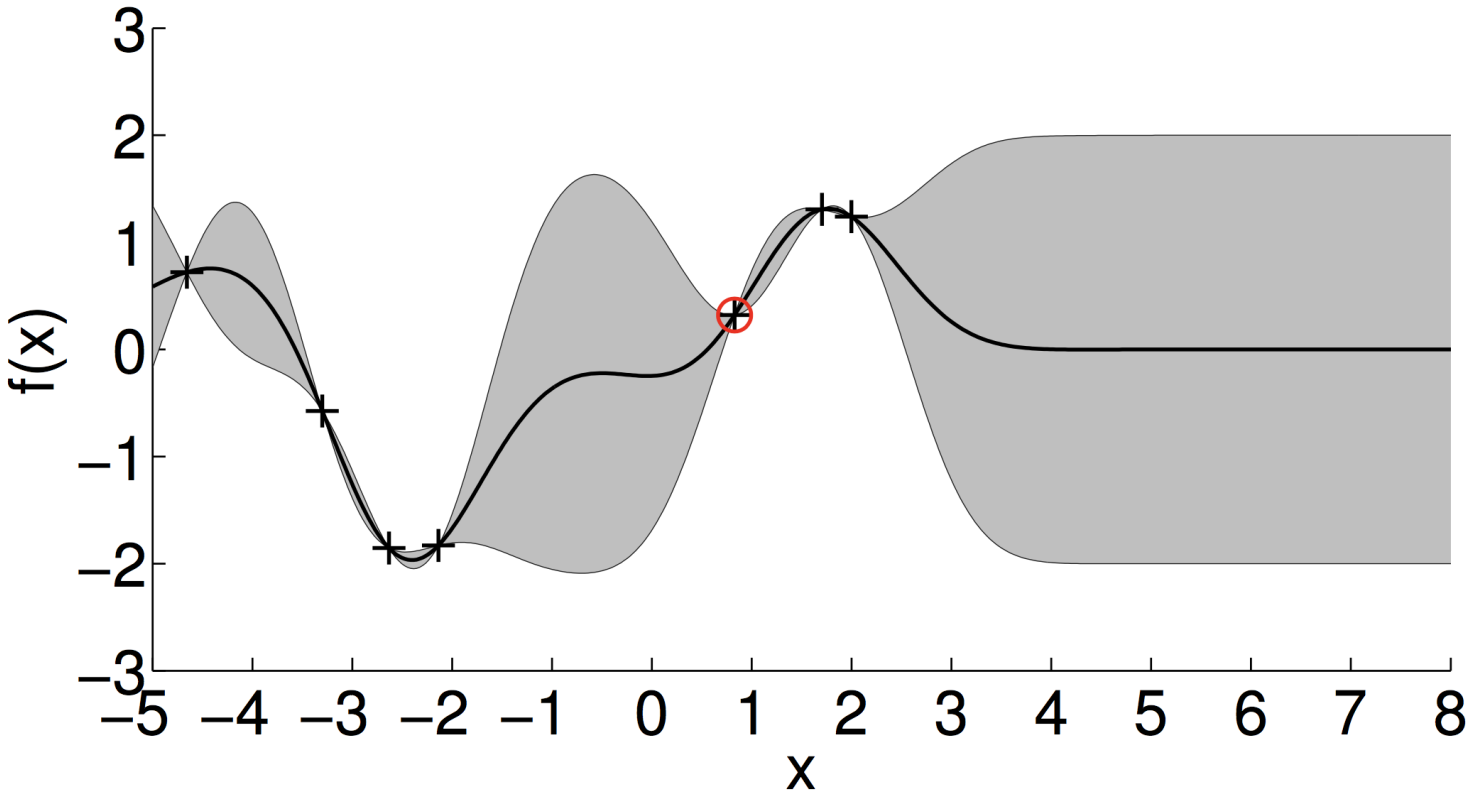


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

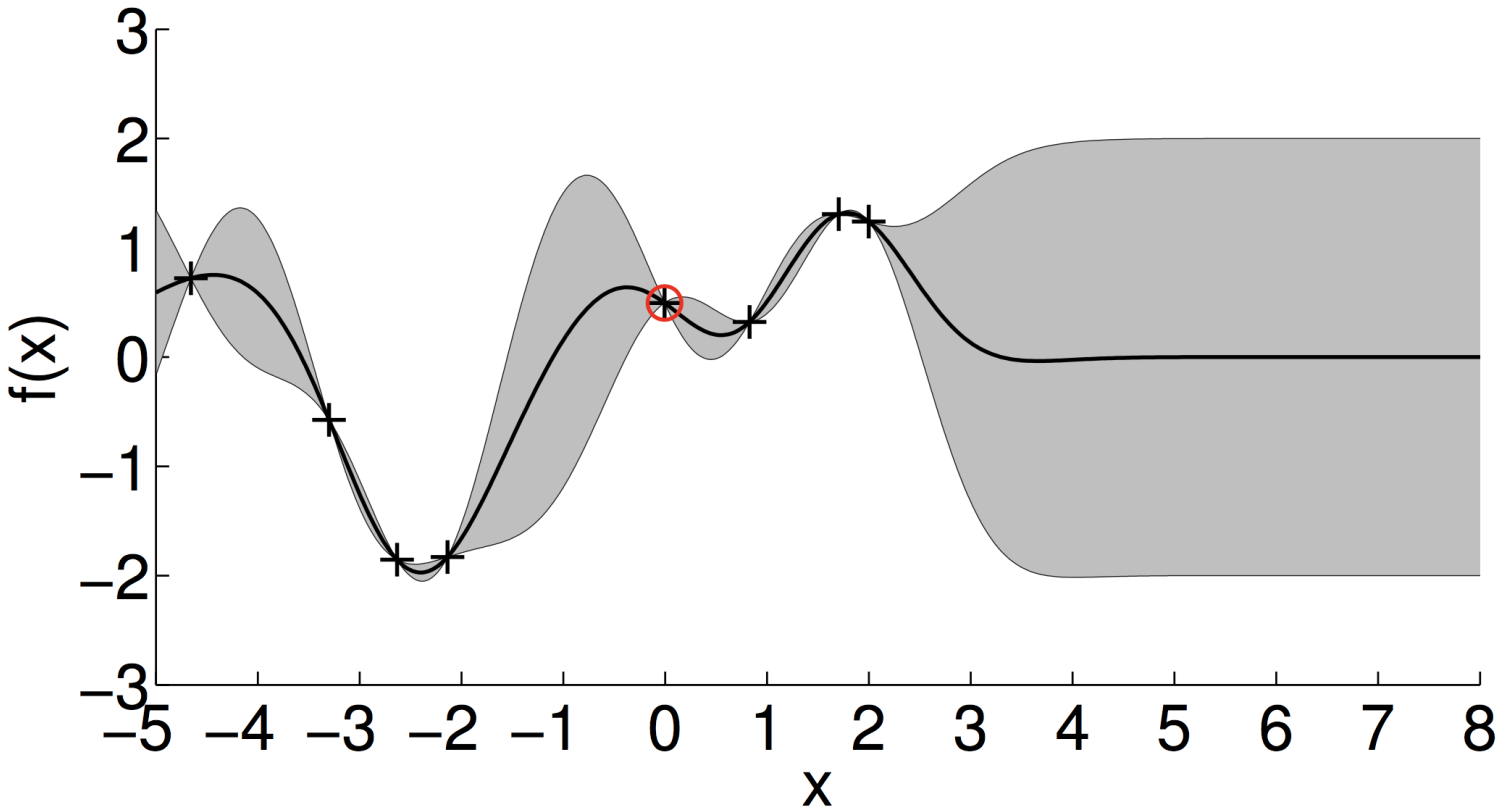


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

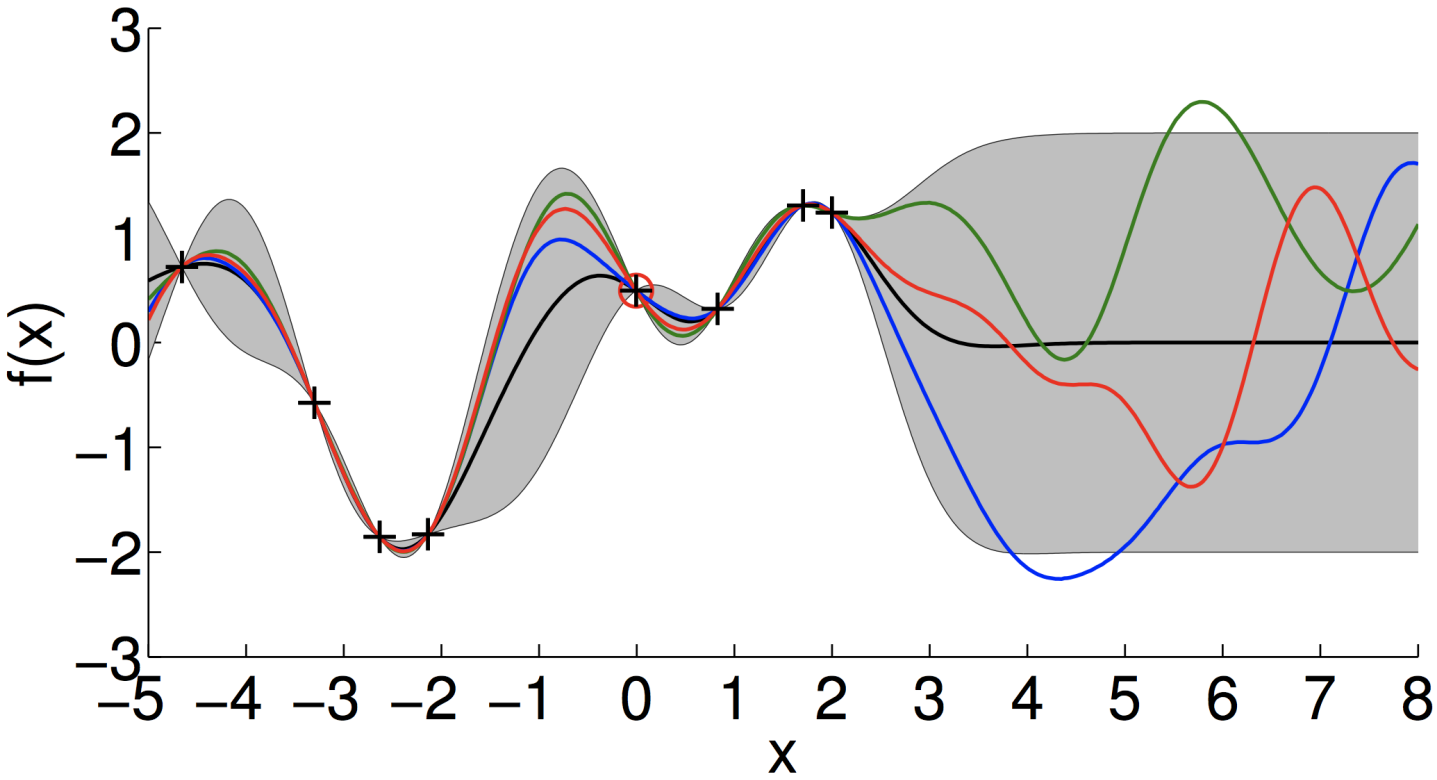


Figure – **Posterior** belief about the function

Predictive (marginal) mean and variance

$$\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \mu_*(\mathbf{x}_*) = \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{y}$$

$$\mathrm{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] = \sigma_*^2(\mathbf{x}_*) = K_{**} - \mathbf{k}_*^\top [\mathbf{K} + \sigma^2 \mathrm{I}_m]^{-1} \mathbf{k}_*$$

Methodology to perform global optimization of multimodal black-box functions

1. Choose some prior measure over the space of possible objectives f
2. Combine prior and the likelihood to get a posterior over the objective given some observations
3. Use the posterior to decide where to take the next evaluation according to some acquisition function
4. Augment the data set
5. Iterate between 2 and 4 until the evaluation budget is over

Comment: BO can be theoretically formalized in the framework of dynamic programming principle

- Use GP $\mathcal{GP}(\cdot|\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ as a prior for $f(\cdot)$
- GP has marginal closed-form for the posterior mean $\mu_*(\mathbf{x})$ and variance $\sigma_*^2(\mathbf{x}) \Rightarrow$ efficient calculation of acquisition function
  - **Exploration**: Evaluate in places where the variance is large
  - **Exploitation**: Evaluate in places where the mean is low

Acquisition functions balance these two factors to determine where to evaluate next

- BO is an strategy to transform the problem

$$\mathbf{x}_{\min} = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

unsolvable!

into a series of problems

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}|S_t),$$

solvable!

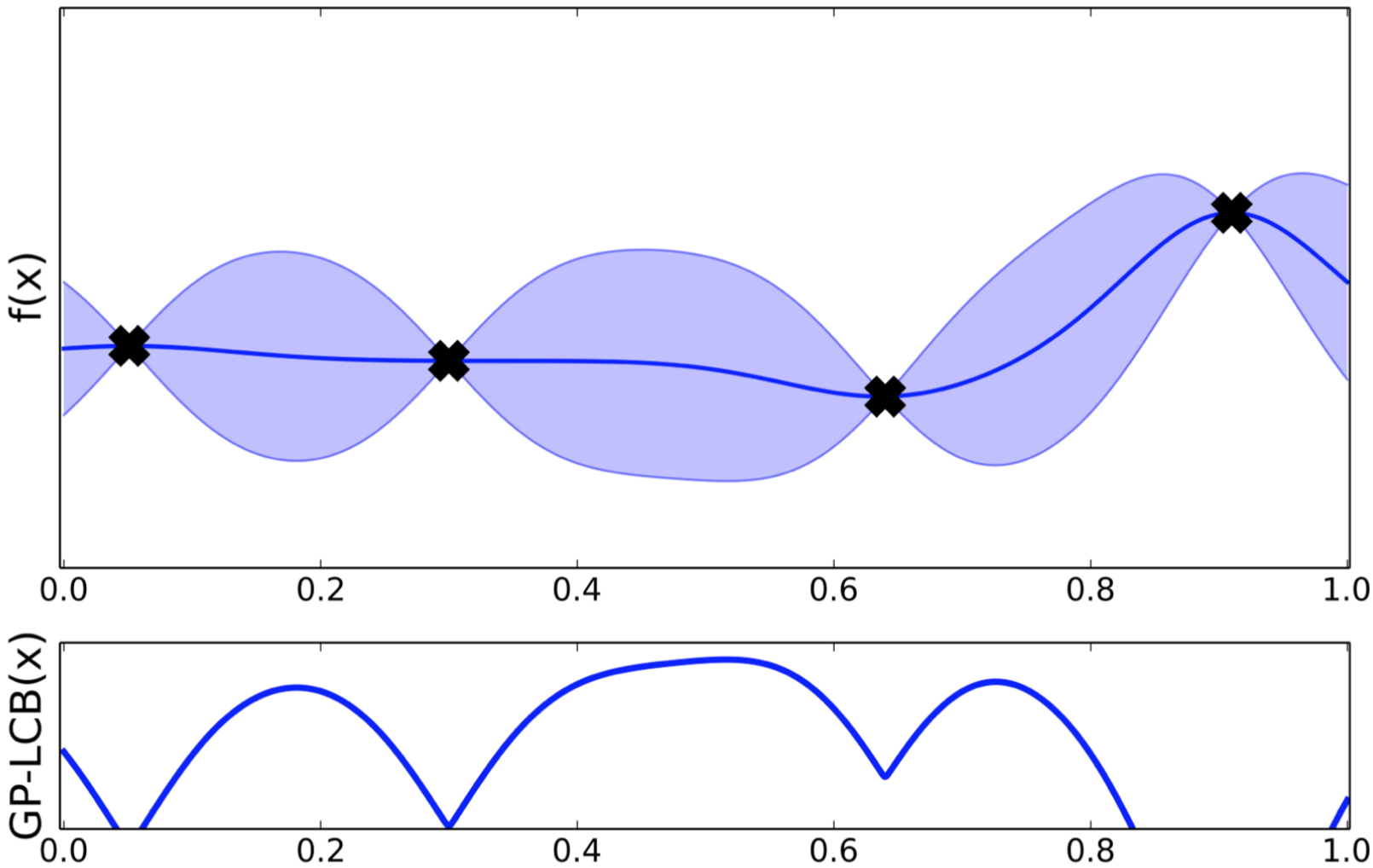
where

  - $\alpha(\mathbf{x})$ is not so expensive to evaluate
  - Gradients of $\alpha(\mathbf{x})$ are typically available
  - Still need to find $\mathbf{x}_{t+1}$: DIRECT, gradient methods, SA

GP Upper (lower) Confidence Band

Direct balance between exploration and exploitation (ζ is a user-defined parameter):

$$\alpha_{LCB}(\mathbf{x}) = -\mu_*(\mathbf{x}) + \zeta \cdot \sigma_*(\mathbf{x})$$

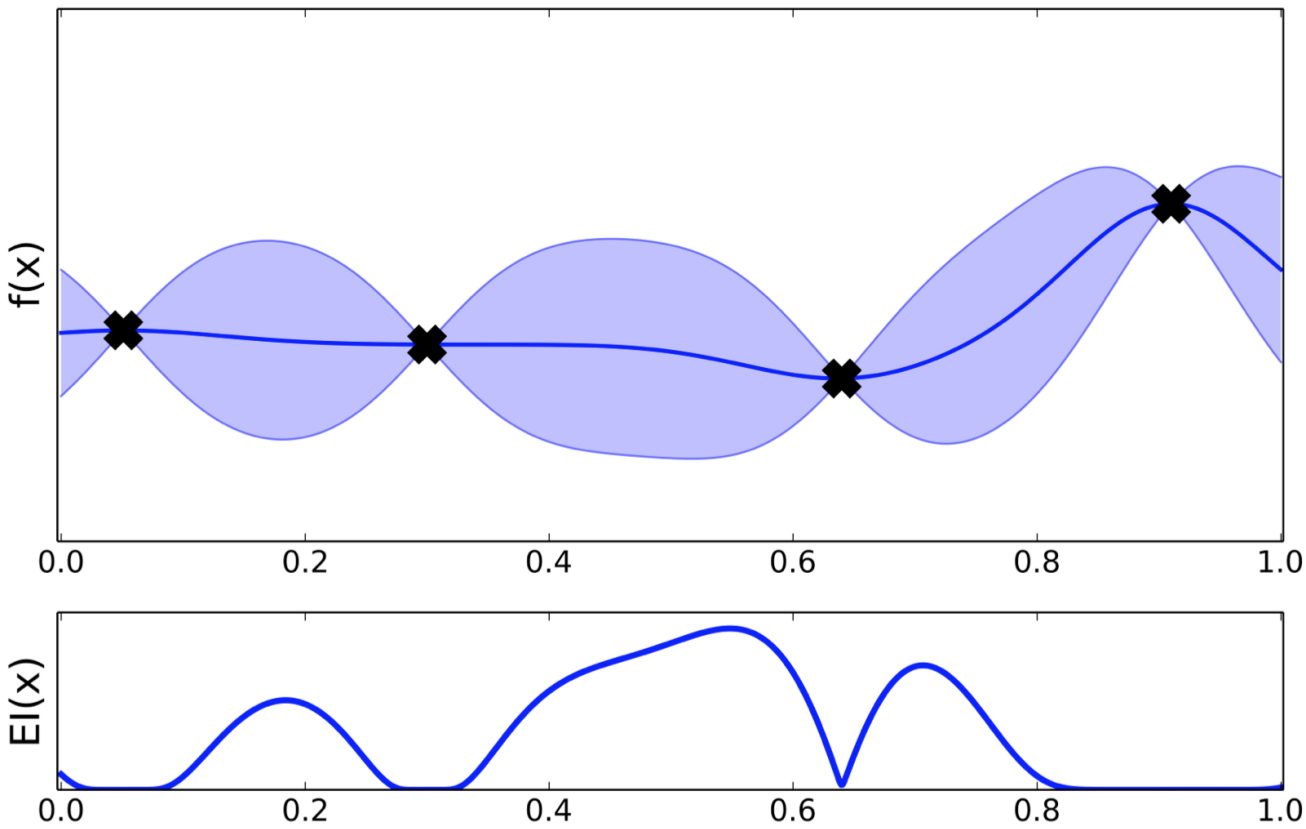


Expected Improvement

Let us denote by $\Delta(\mathbf{x}) = y_{\text{best}} - \mu_*(\mathbf{x})$, then

$$\alpha_{EI}(\mathbf{x}) = \int_y \max(0, y_{\text{best}} - y_*) p(y_*|\mathbf{x}) dy_* =$$

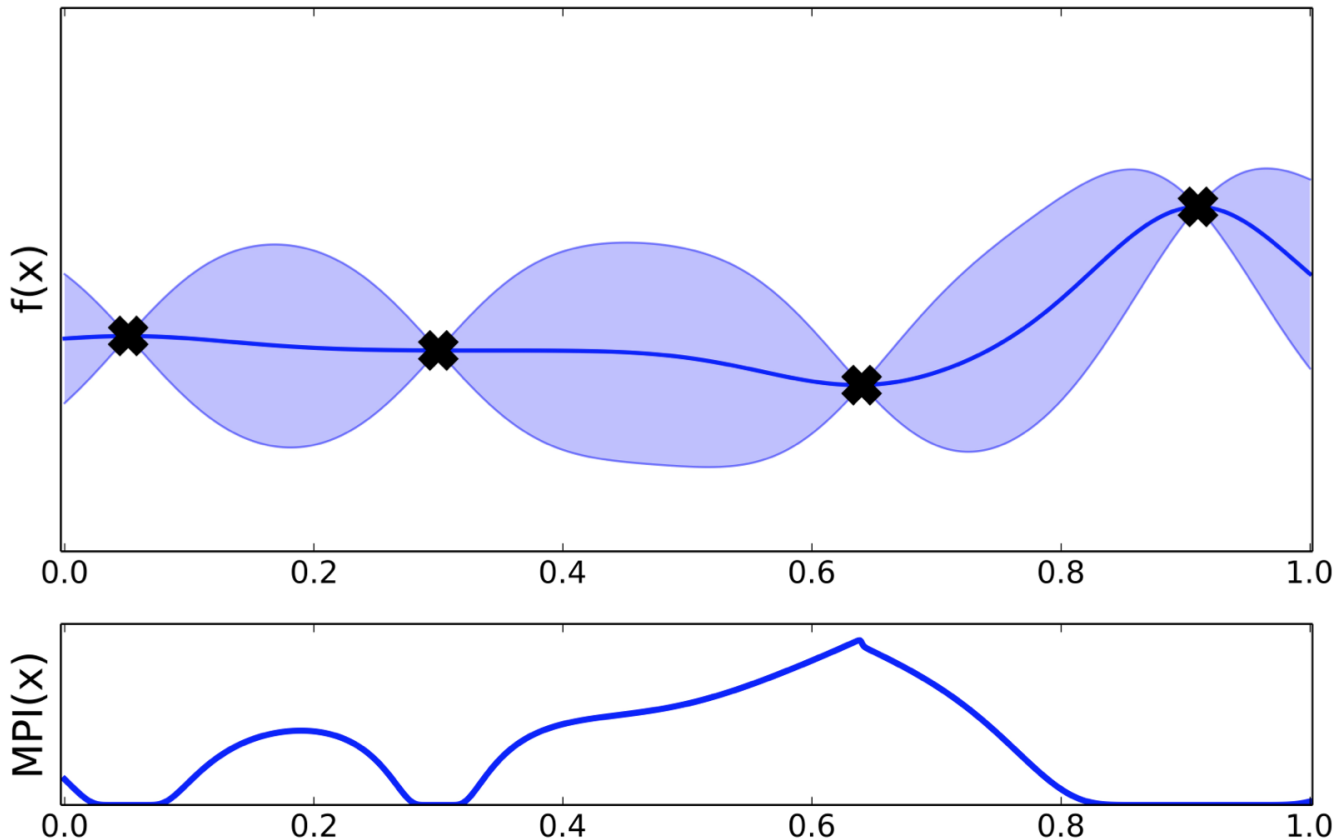
$$= \max(0, \Delta(\mathbf{x})) - \sigma_*(\mathbf{x}) \varphi\left(\frac{\Delta(\mathbf{x})}{\sigma_*(\mathbf{x})}\right) + |\Delta(\mathbf{x})| \Phi\left(-\frac{|\Delta(\mathbf{x})|}{\sigma_*(\mathbf{x})}\right)$$



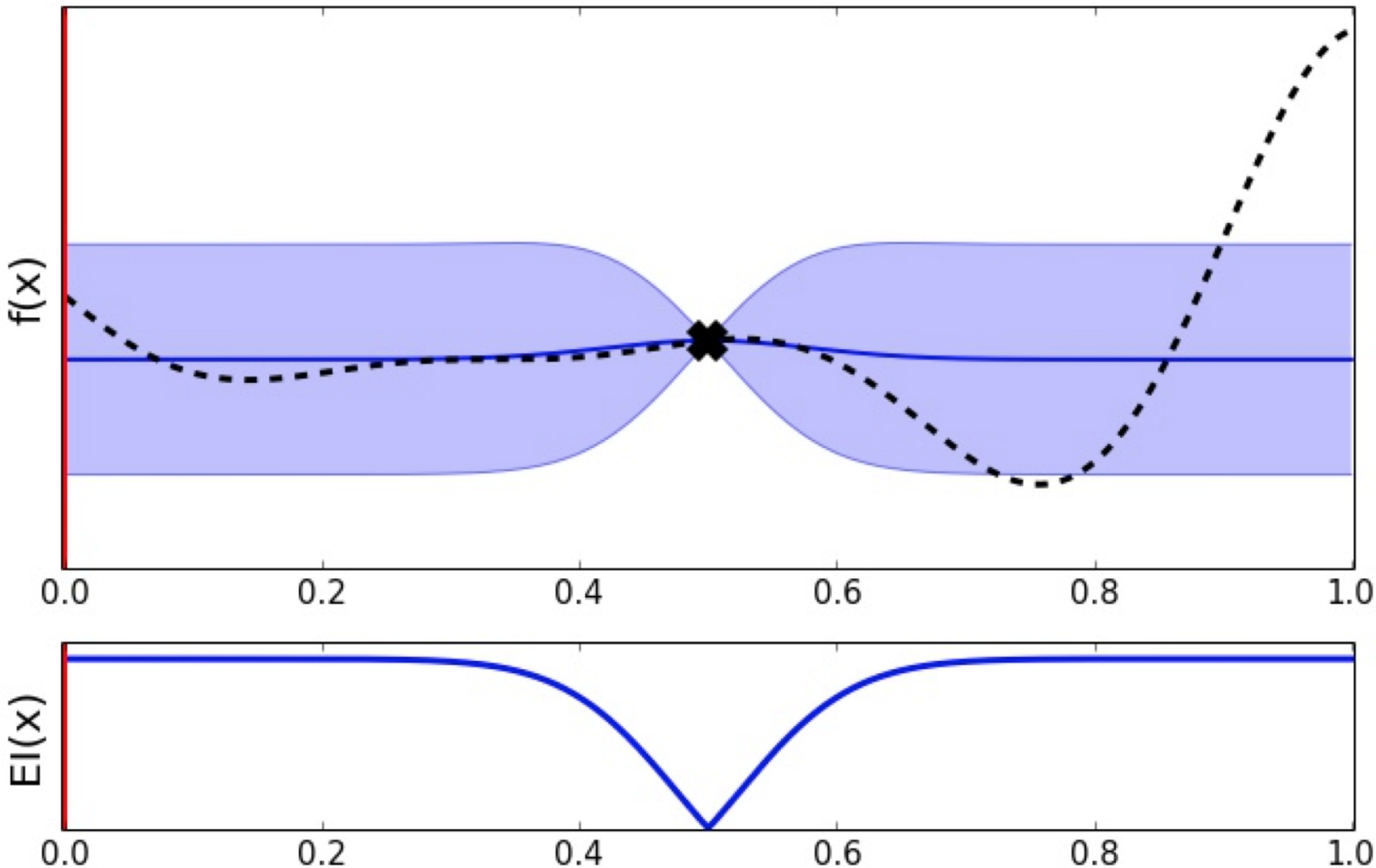
Maximum Probability of Improvement

$$\gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}) - y_{\text{best}}}{\sigma(\mathbf{x})}$$

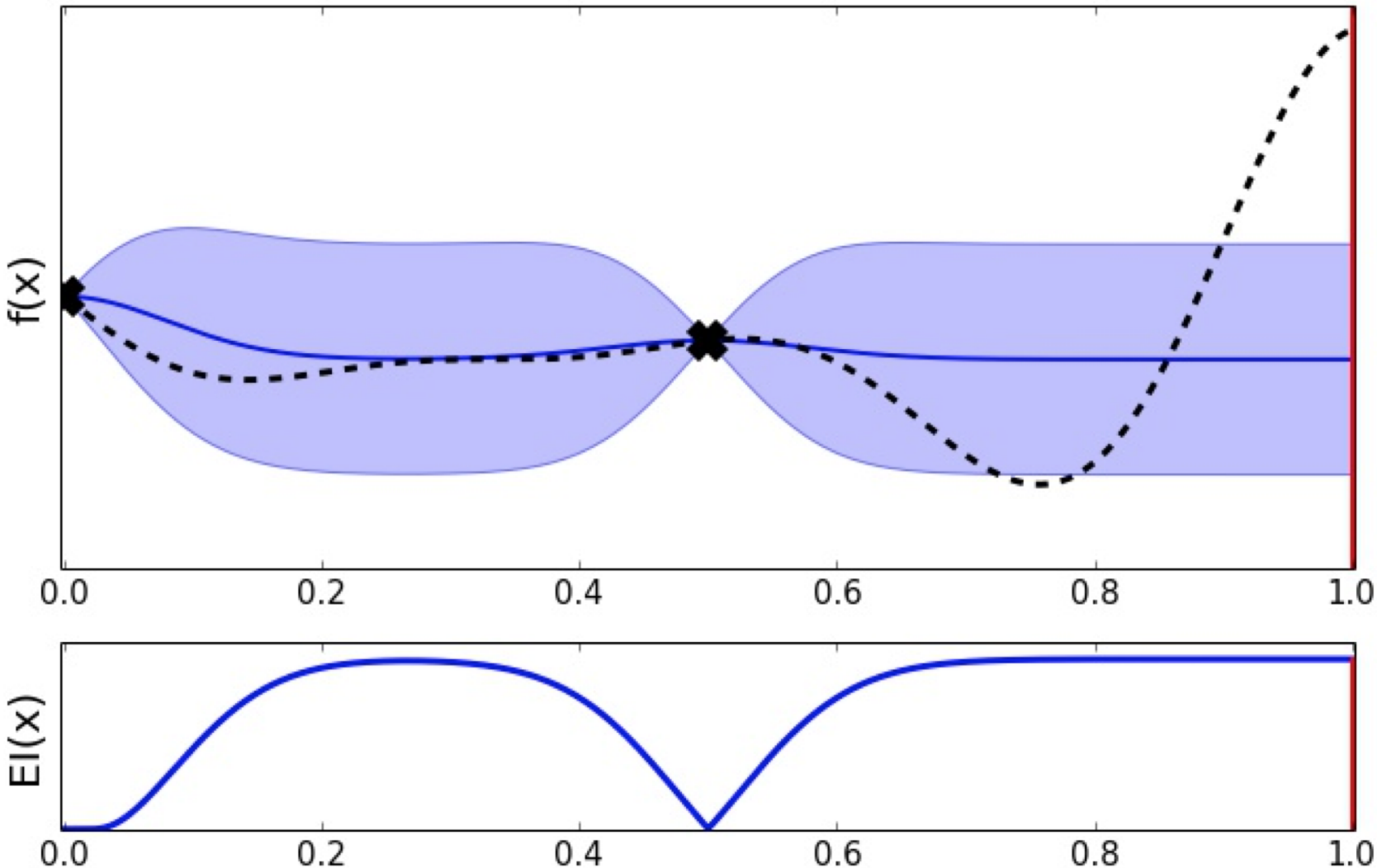
$$\alpha_{MPI}(\mathbf{x}) = \mathbb{P}(f(\mathbf{x}) < y_{\text{best}}) = \Phi(\gamma(\mathbf{x}))$$



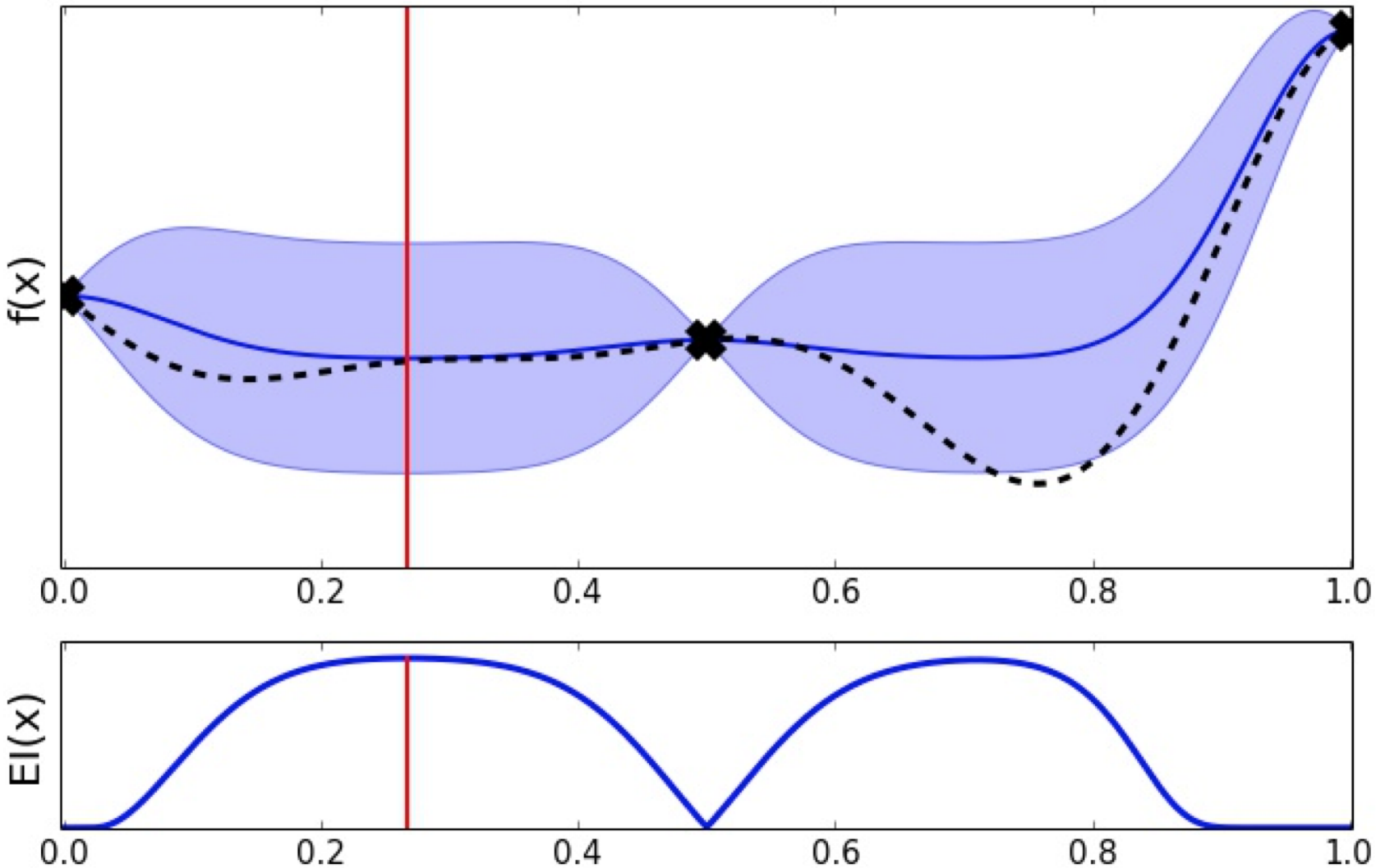
Expected Improvement: Toy Problem



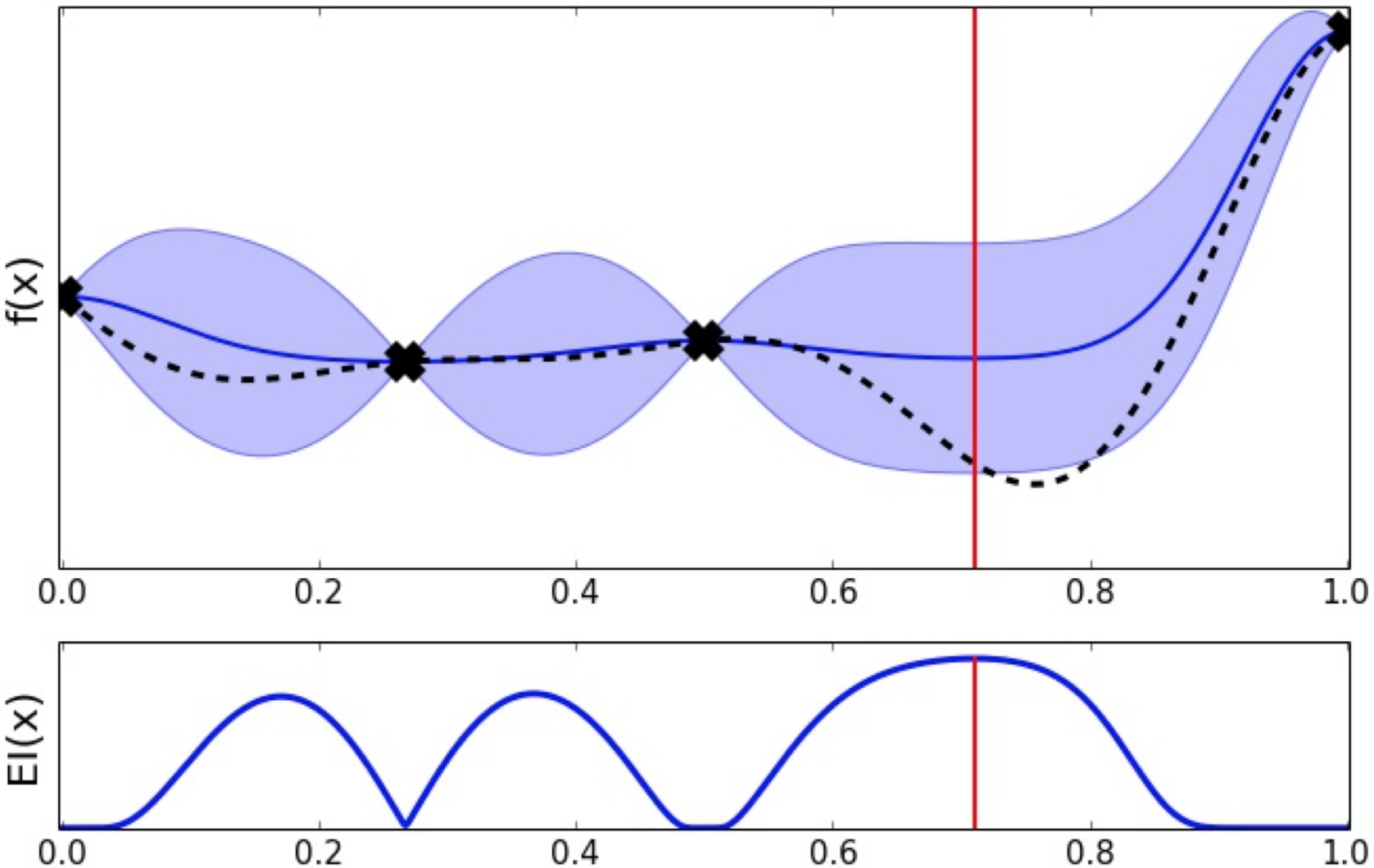
Expected Improvement: Toy Problem



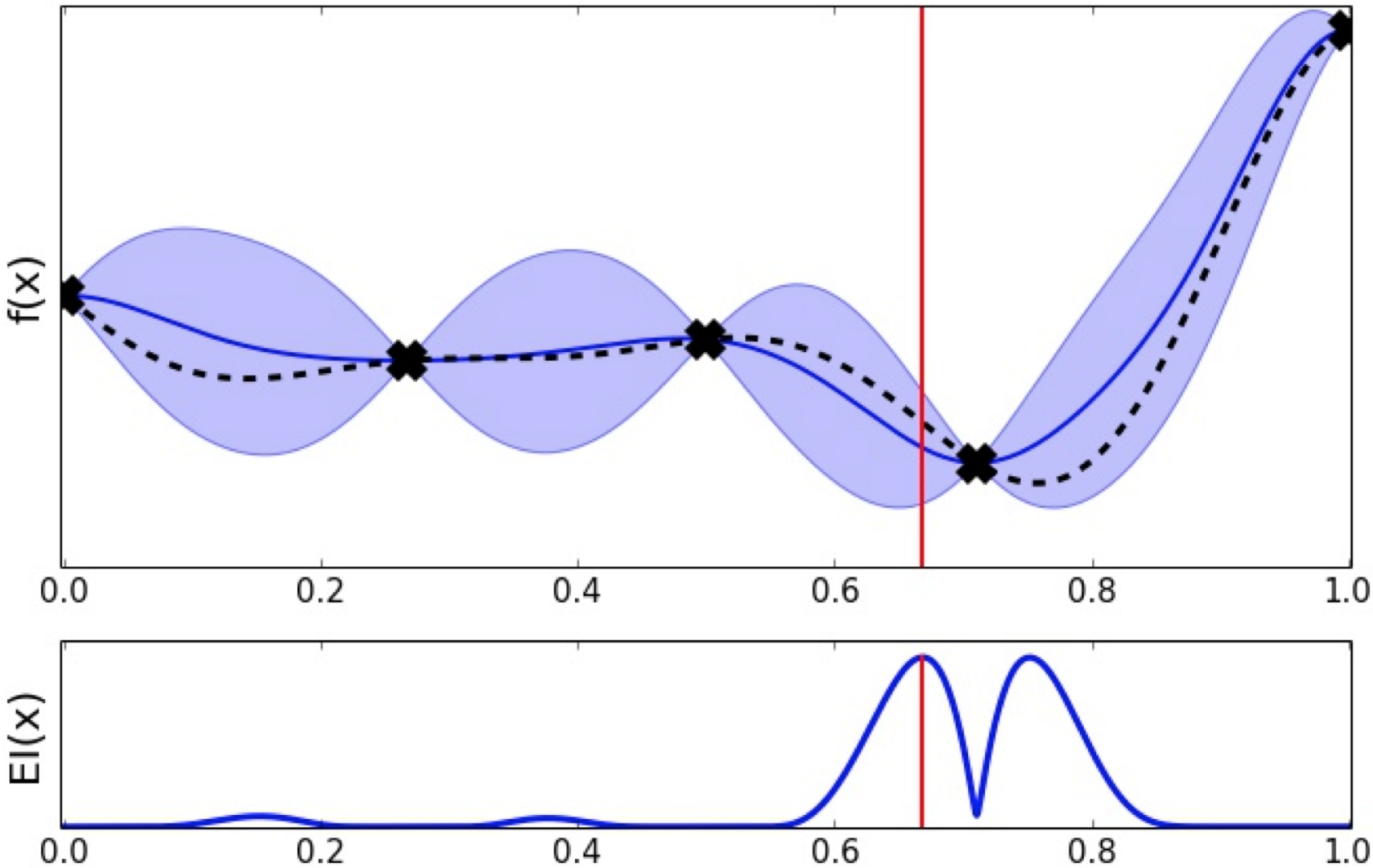
Expected Improvement: Toy Problem



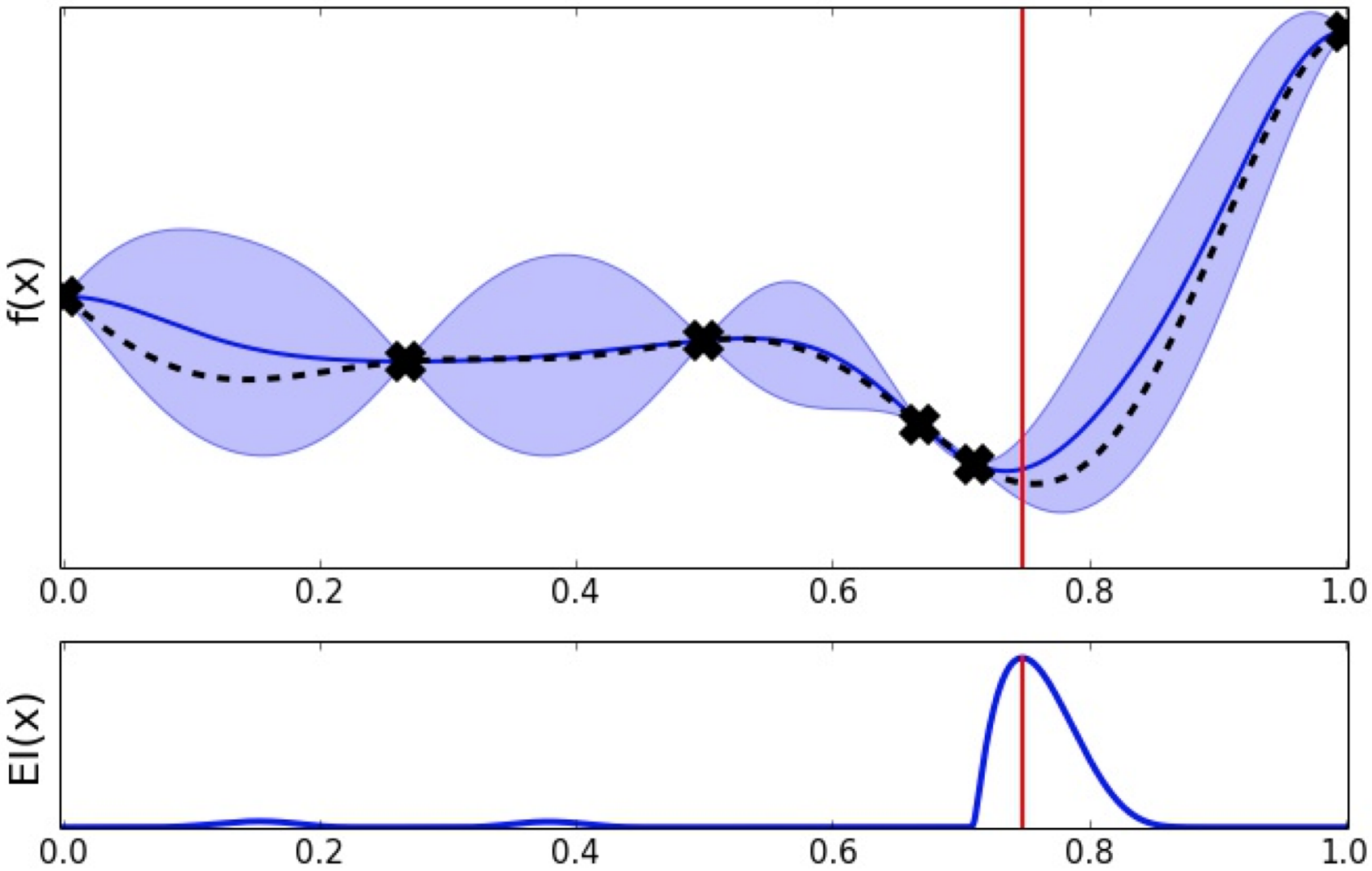
Expected Improvement: Toy Problem



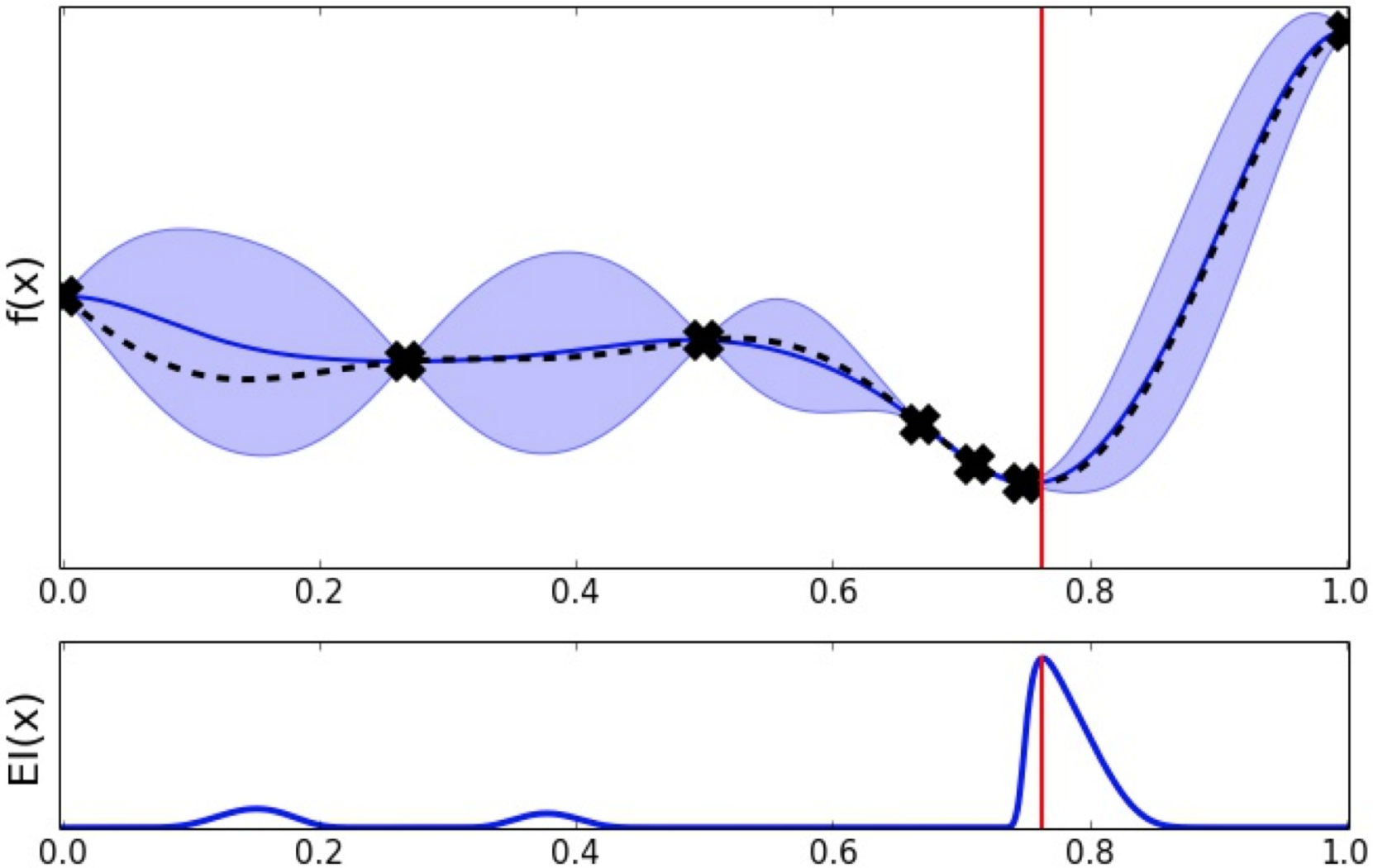
Expected Improvement: Toy Problem



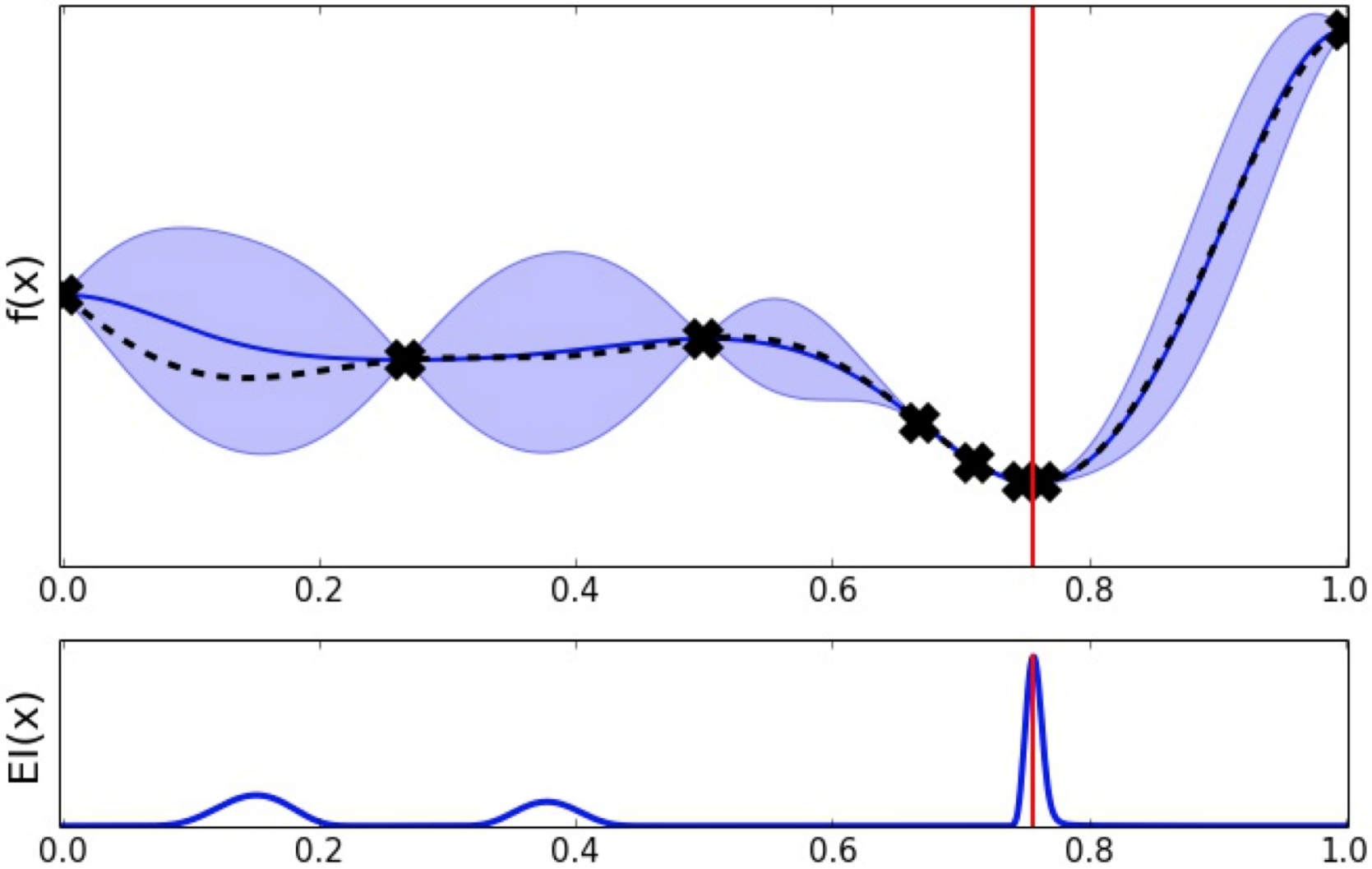
Expected Improvement: Toy Problem



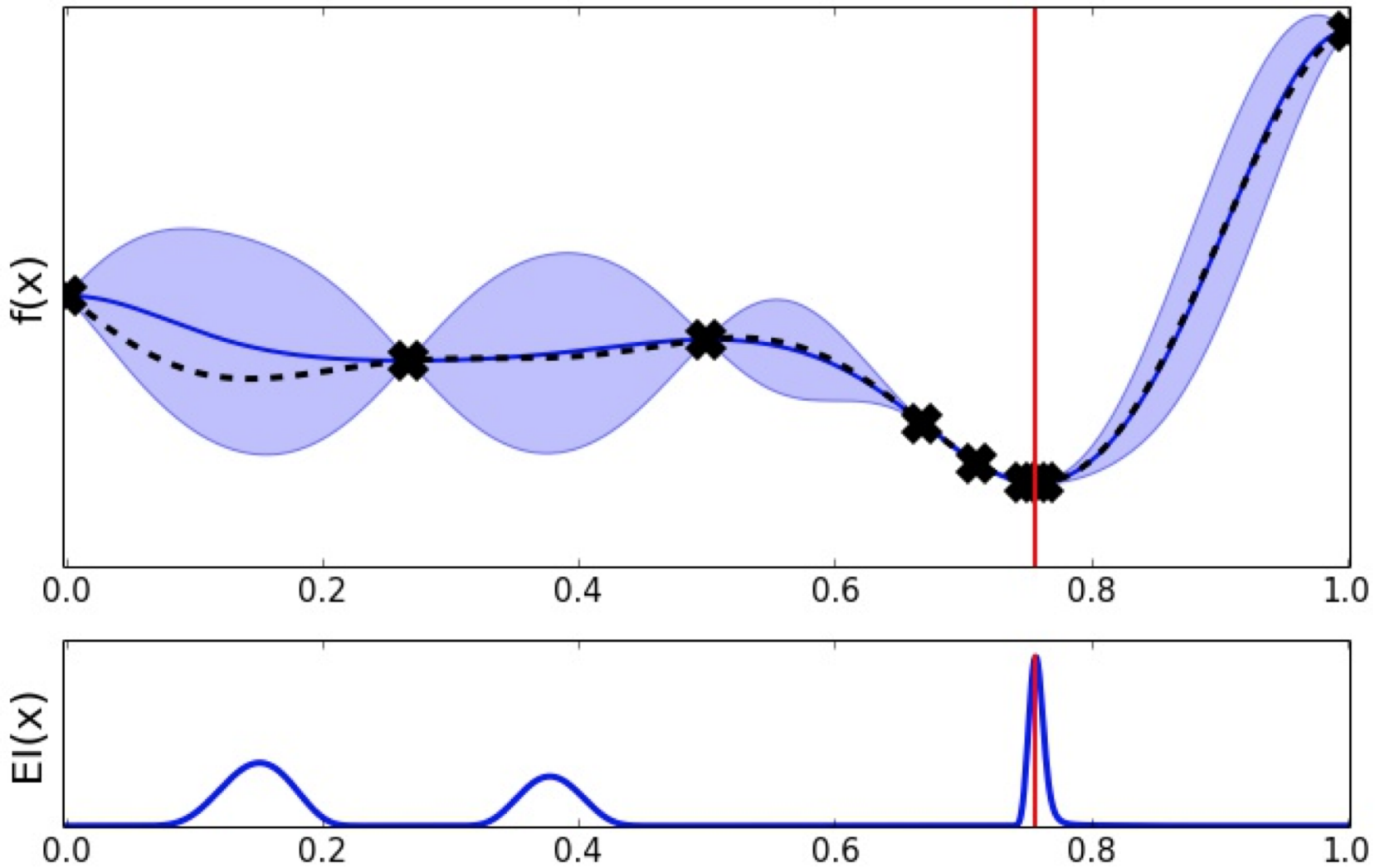
Expected Improvement: Toy Problem



Expected Improvement: Toy Problem



Expected Improvement: Toy Problem



AutoML example: SVM hyperparameters tuning

- Scikit-learn dummy data for testing classifiers:

  n_samples=2500, n_features=45, n_informative=15, n_redundant=5

- Optimize w.r.t. $\mathbf{x} = (C, \gamma)$, where C — penalization, and γ — kernel width
- The target function $\mathcal{L}(\mathbf{x})$ is the AUC based on three fold cross-validation

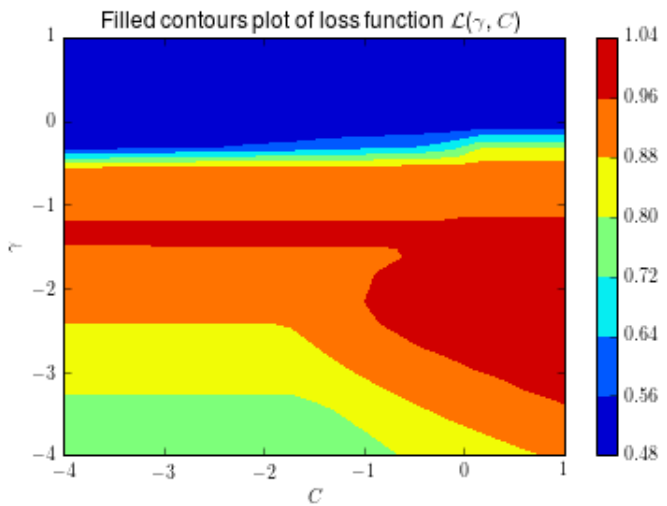
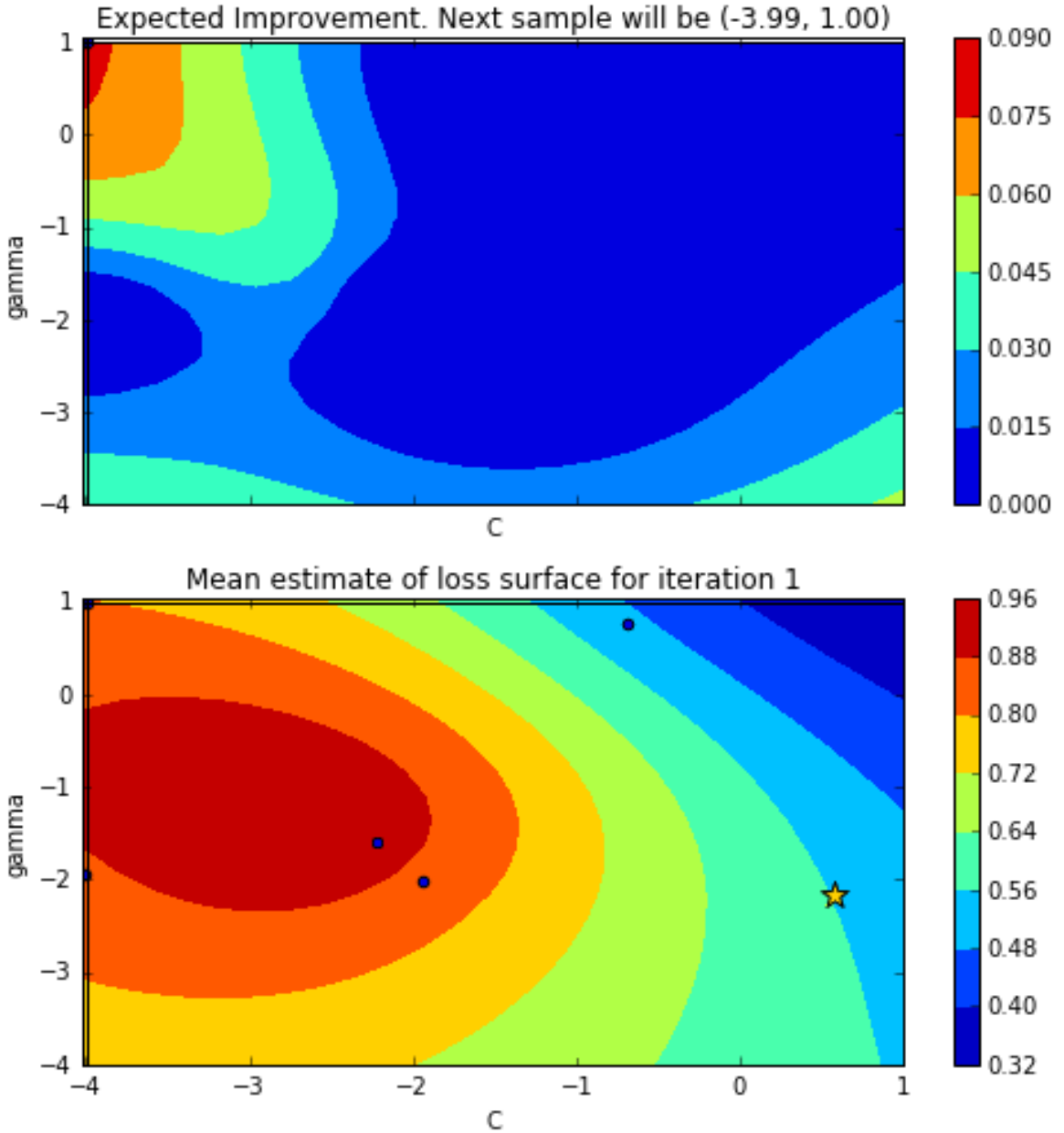
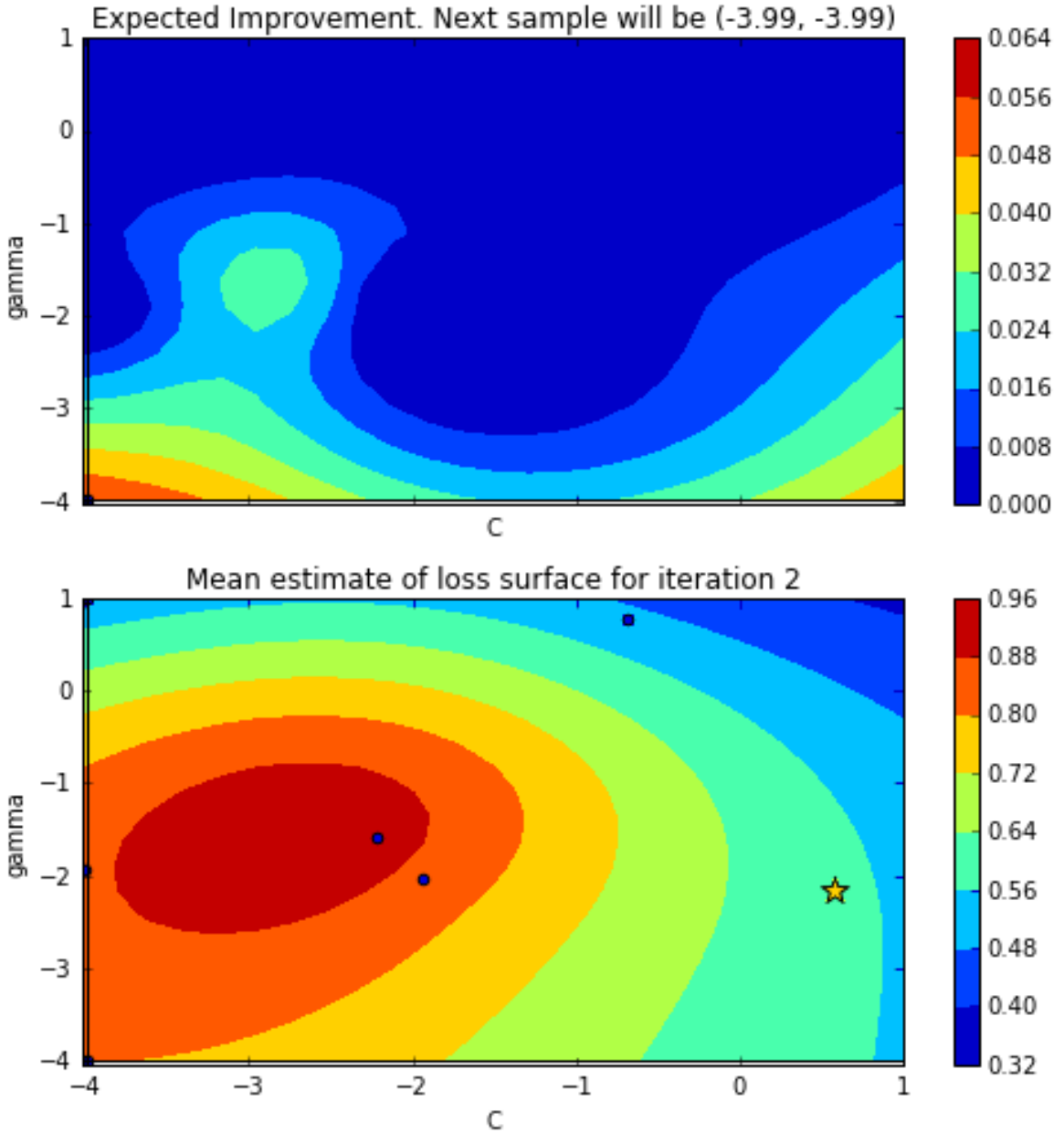


Figure – The target surface w.r.t. $\mathbf{x}$ to see where the true optimum is

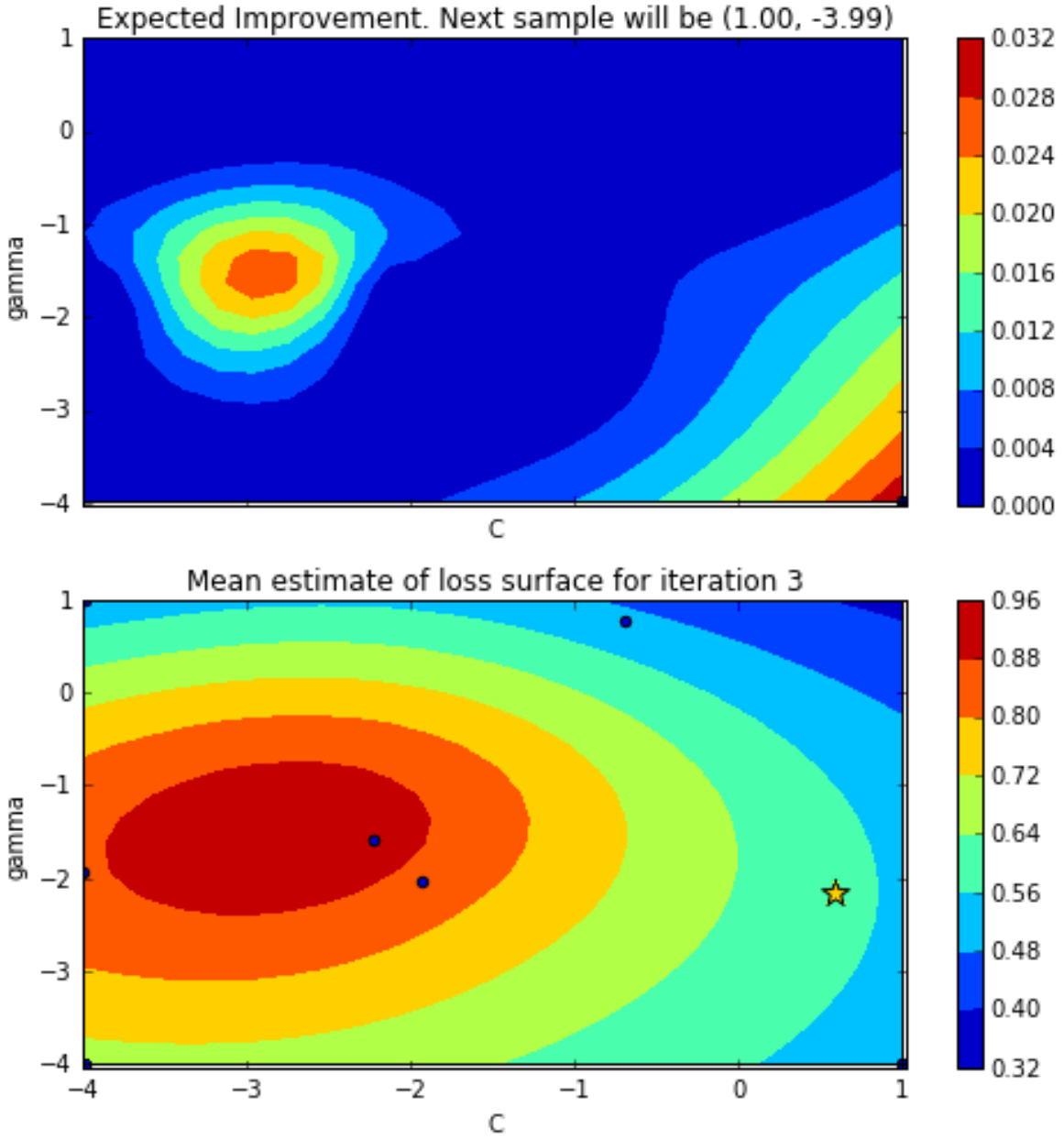
Iterations


Expected Improvement. Next sample will be (-3.99, 1.00)
gamma
C
1
0
−1
−2
−3
−4
0.090
0.075
0.060
0.045
0.030
0.015
0.000
Mean estimate of loss surface for iteration 1
gamma
C
−4
−3
−2
−1
0
1
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40
0.32

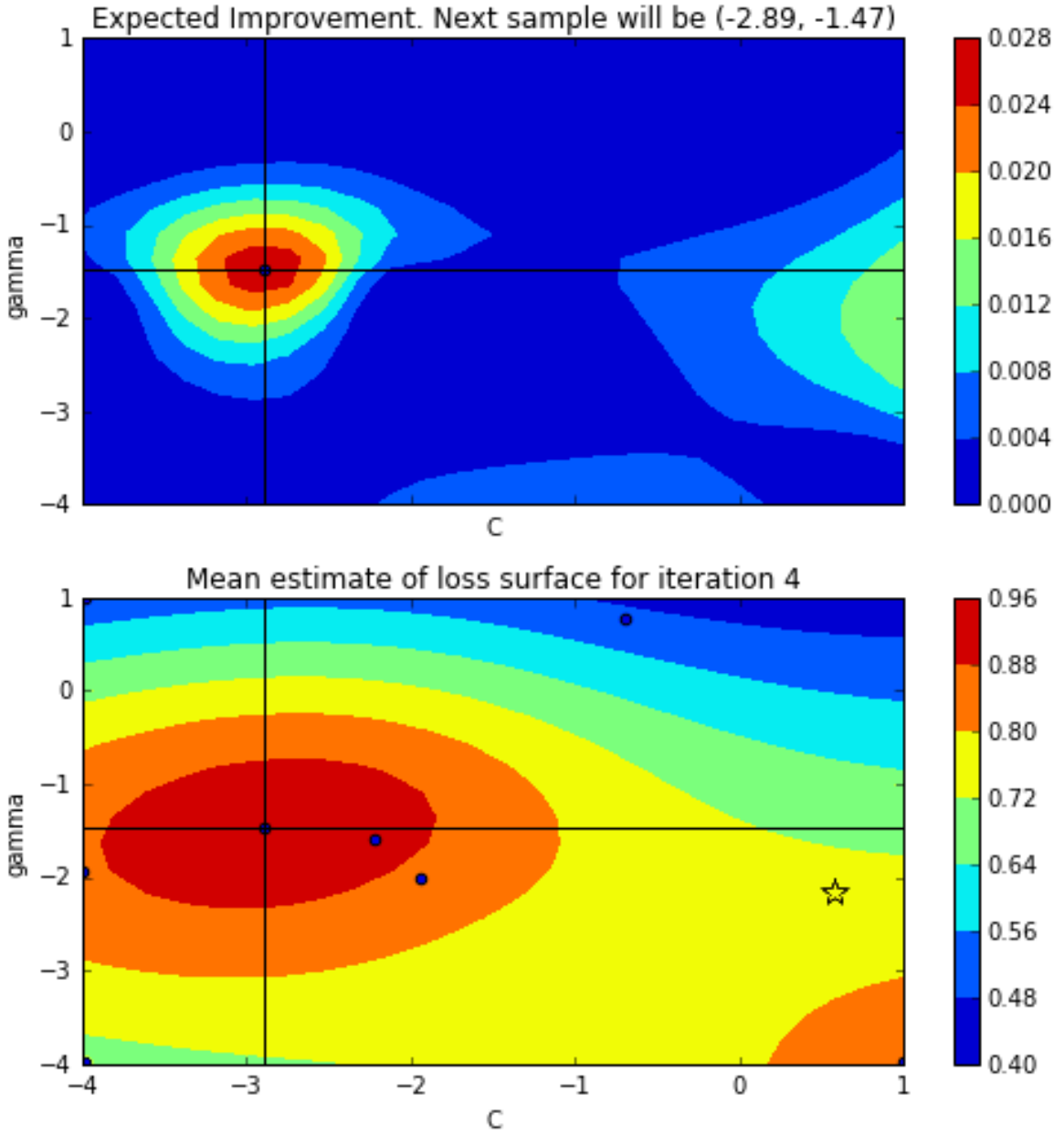
Iterations


Expected Improvement. Next sample will be (-3.99, -3.99)
gamma
C
0.064
0.056
0.048
0.040
0.032
0.024
0.016
0.008
0.000
Mean estimate of loss surface for iteration 2
gamma
C
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40
0.32

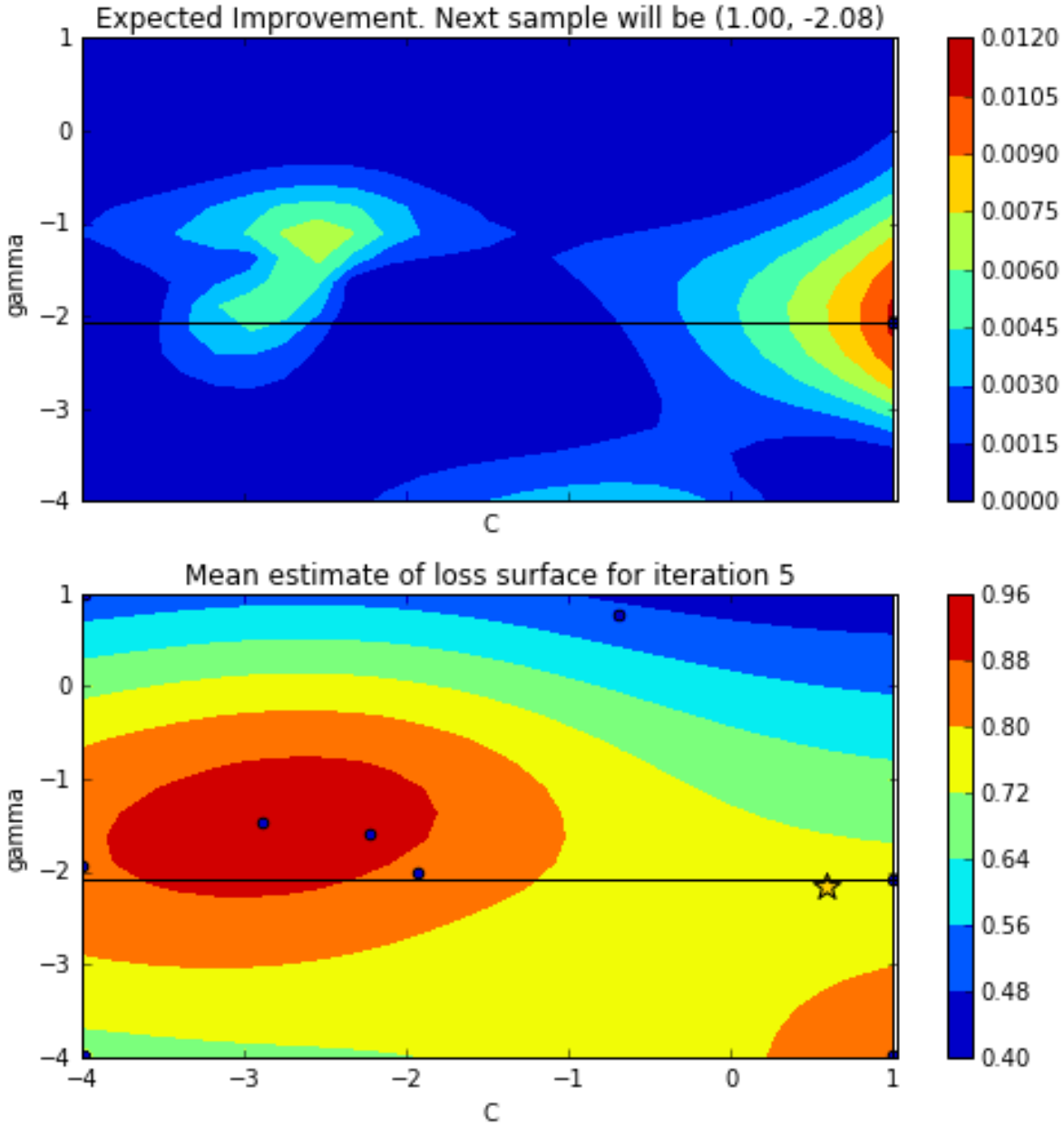
Iterations


Expected Improvement. Next sample will be (1.00, -3.99)
gamma
C
0.032
0.028
0.024
0.020
0.016
0.012
0.008
0.004
0.000
Mean estimate of loss surface for iteration 3
gamma
C
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40
0.32

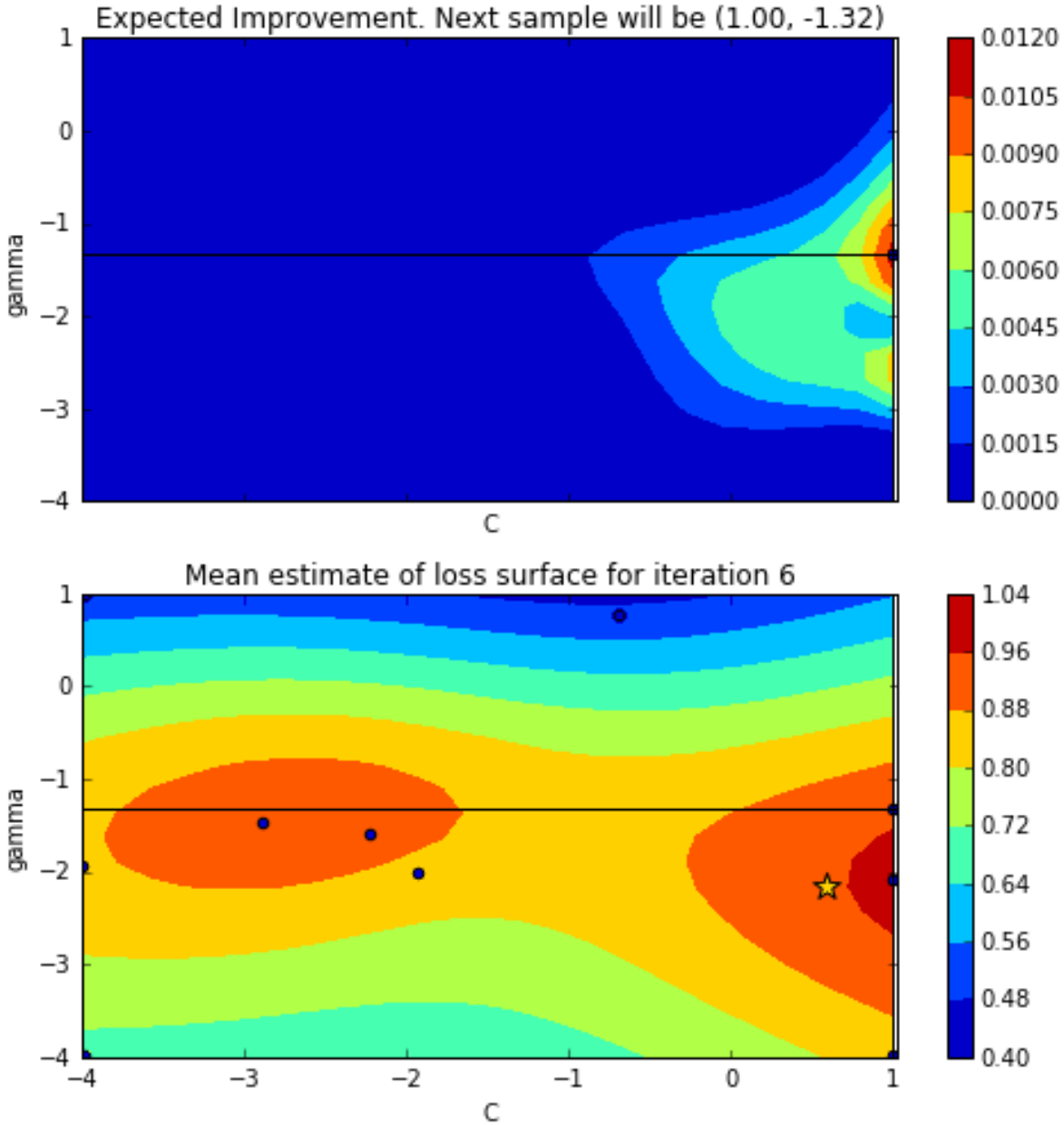
Iterations


Expected Improvement. Next sample will be (-2.89, -1.47)
gamma
C
0.028
0.024
0.020
0.016
0.012
0.008
0.004
0.000
Mean estimate of loss surface for iteration 4
gamma
C
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

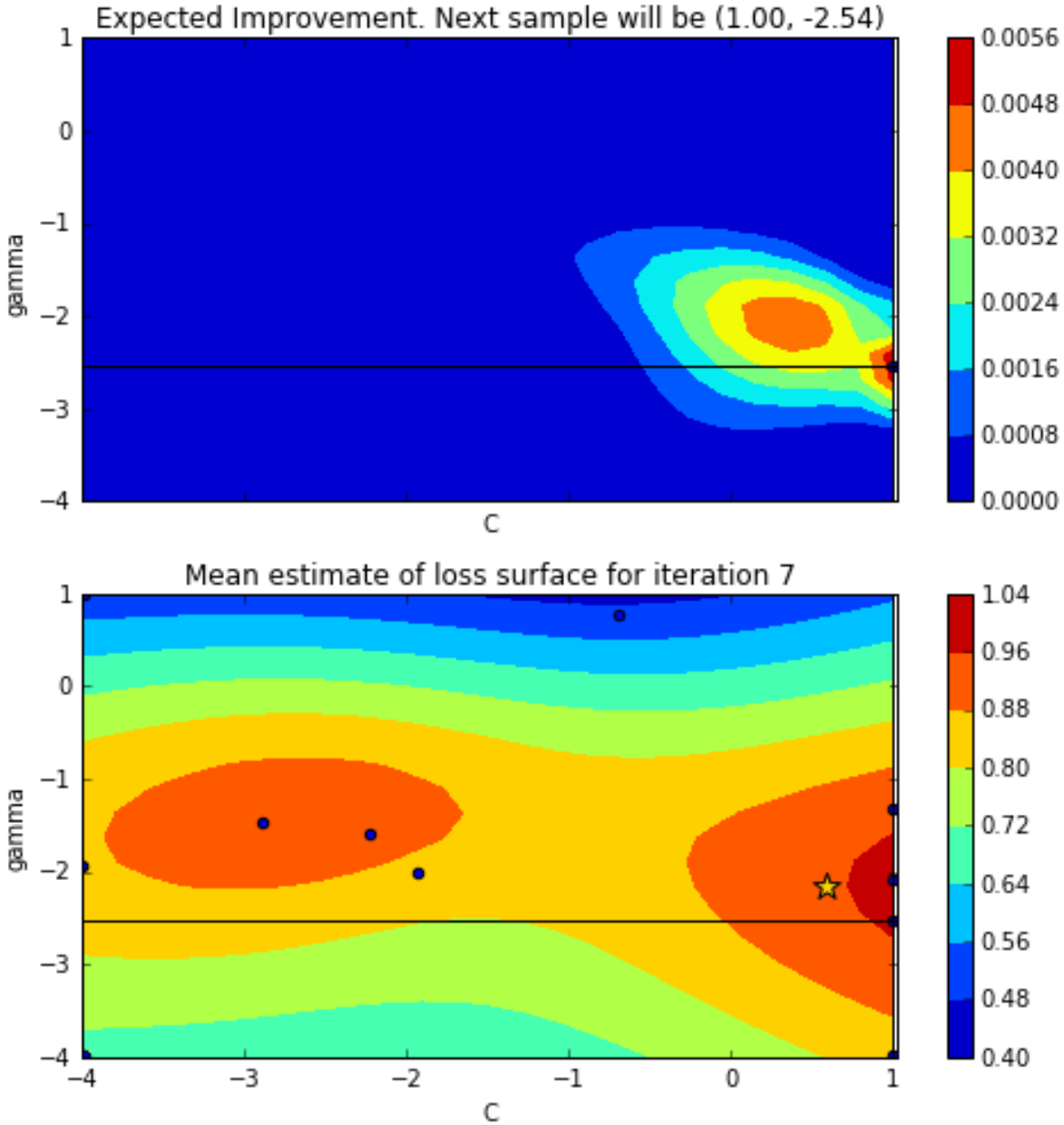
Iterations


Expected Improvement. Next sample will be (1.00, -2.08)
gamma
C
0.0120
0.0105
0.0090
0.0075
0.0060
0.0045
0.0030
0.0015
0.0000
Mean estimate of loss surface for iteration 5
gamma
C
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

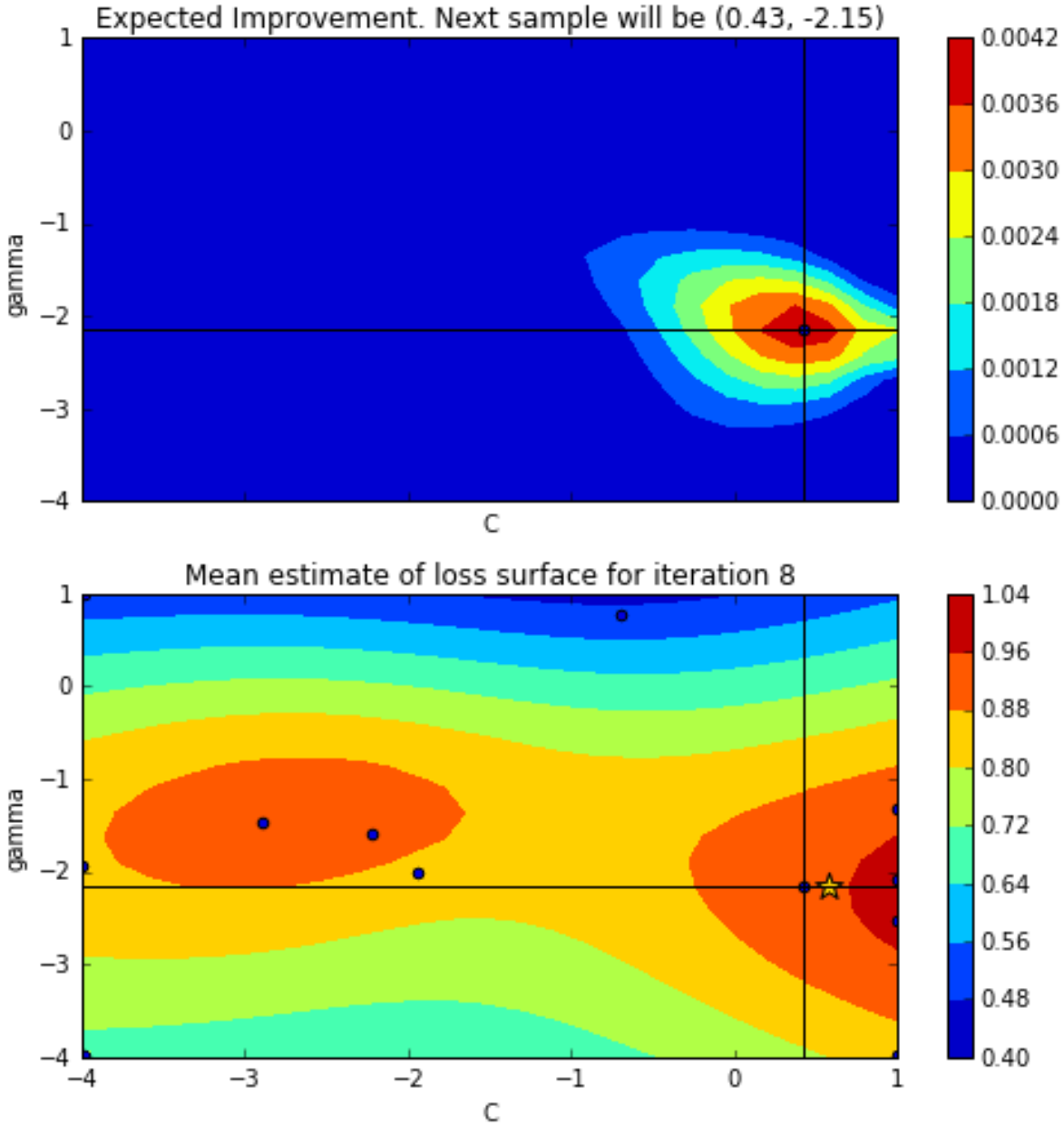
Iterations


Expected Improvement. Next sample will be (1.00, -1.32)
gamma
C
0.0120
0.0105
0.0090
0.0075
0.0060
0.0045
0.0030
0.0015
0.0000
Mean estimate of loss surface for iteration 6
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

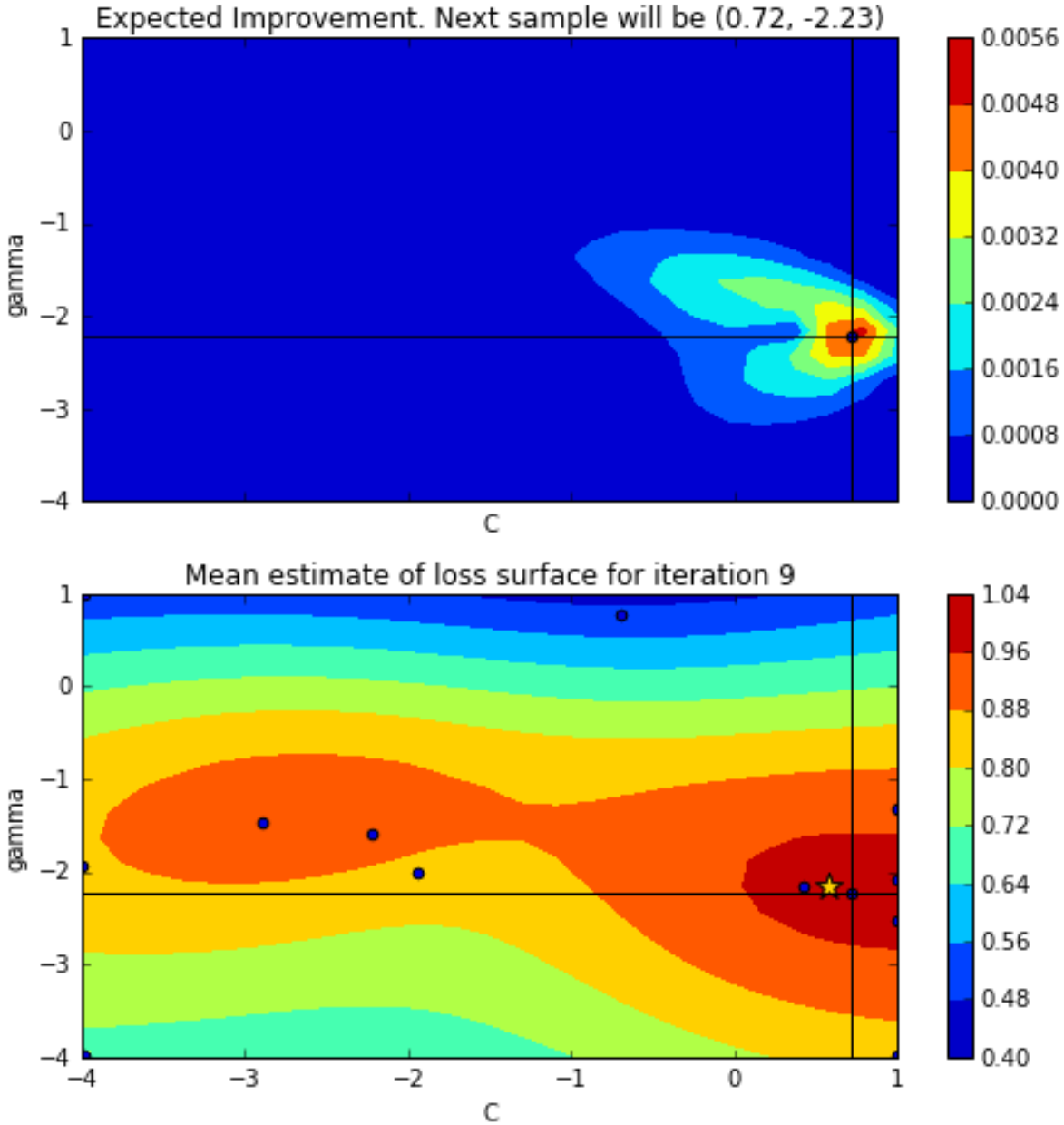
Iterations


Expected Improvement. Next sample will be (1.00, -2.54)
gamma
C
0.0056
0.0048
0.0040
0.0032
0.0024
0.0016
0.0008
0.0000
Mean estimate of loss surface for iteration 7
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

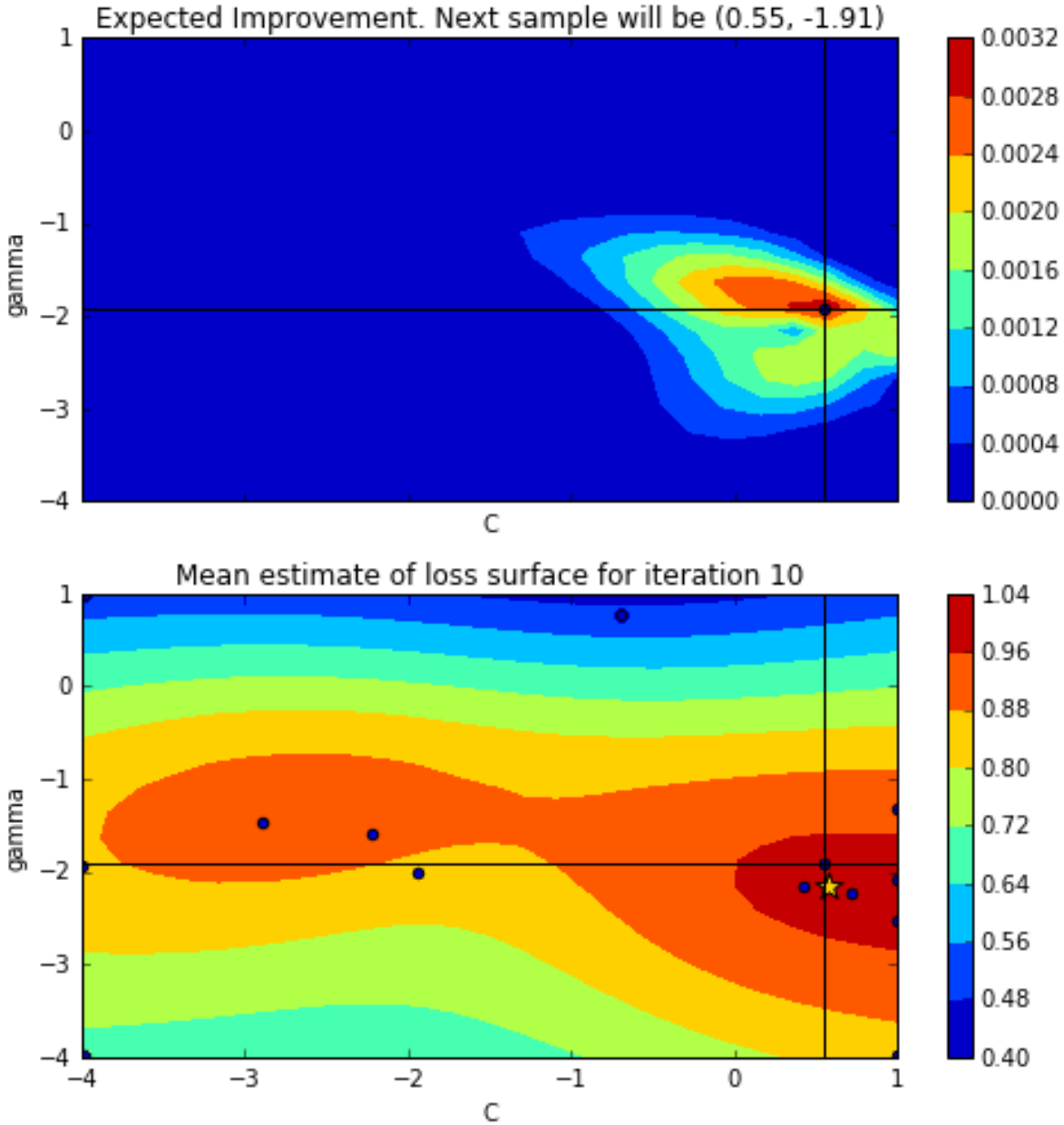
Iterations


Expected Improvement. Next sample will be (0.43, -2.15)
gamma
C
1
0
−1
−2
−3
−4
0.0042
0.0036
0.0030
0.0024
0.0018
0.0012
0.0006
0.0000
Mean estimate of loss surface for iteration 8
gamma
C
−4
−3
−2
−1
0
1
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

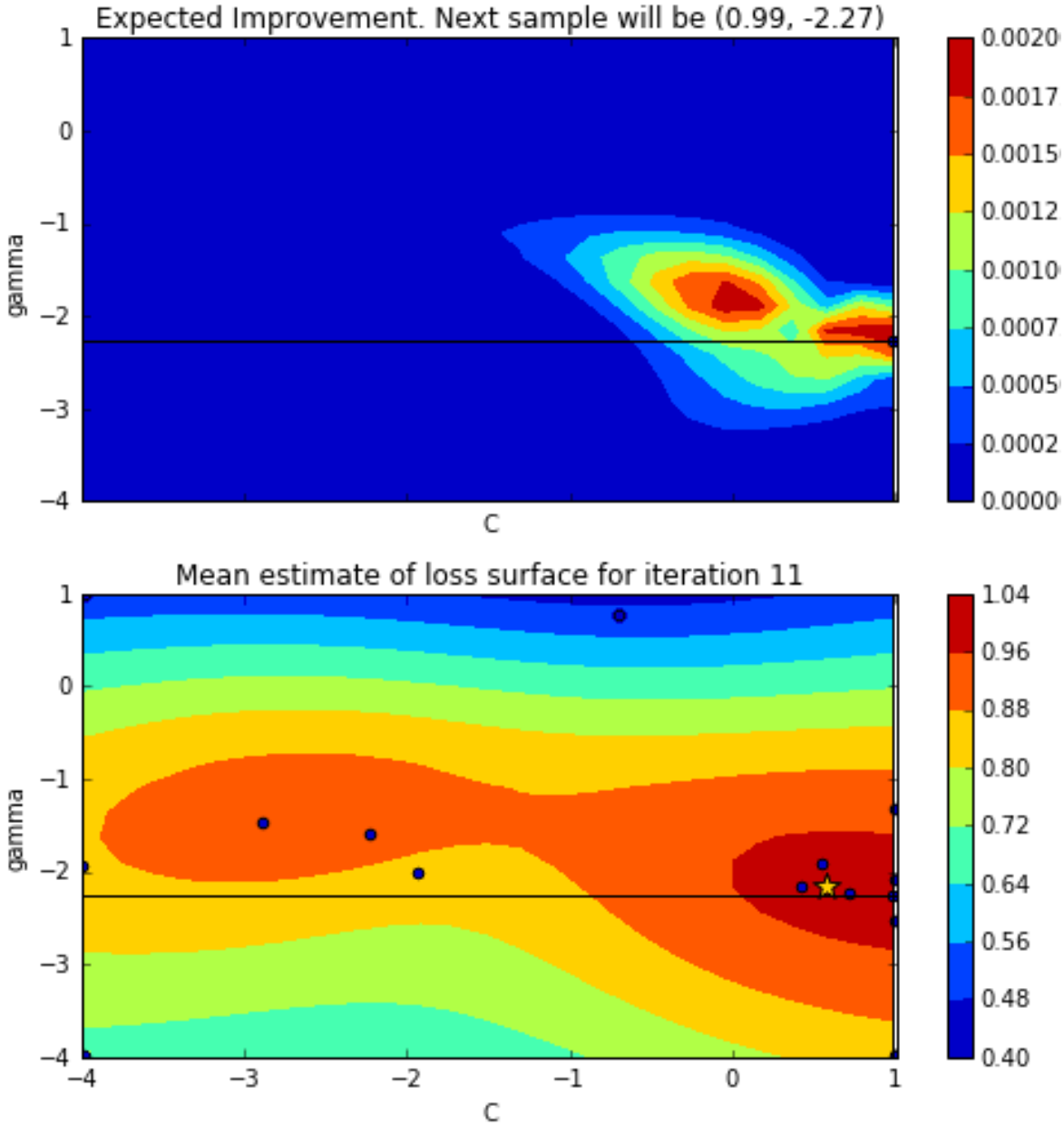
Iterations


Expected Improvement. Next sample will be (0.72, -2.23)
gamma
C
0.0056
0.0048
0.0040
0.0032
0.0024
0.0016
0.0008
0.0000
Mean estimate of loss surface for iteration 9
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

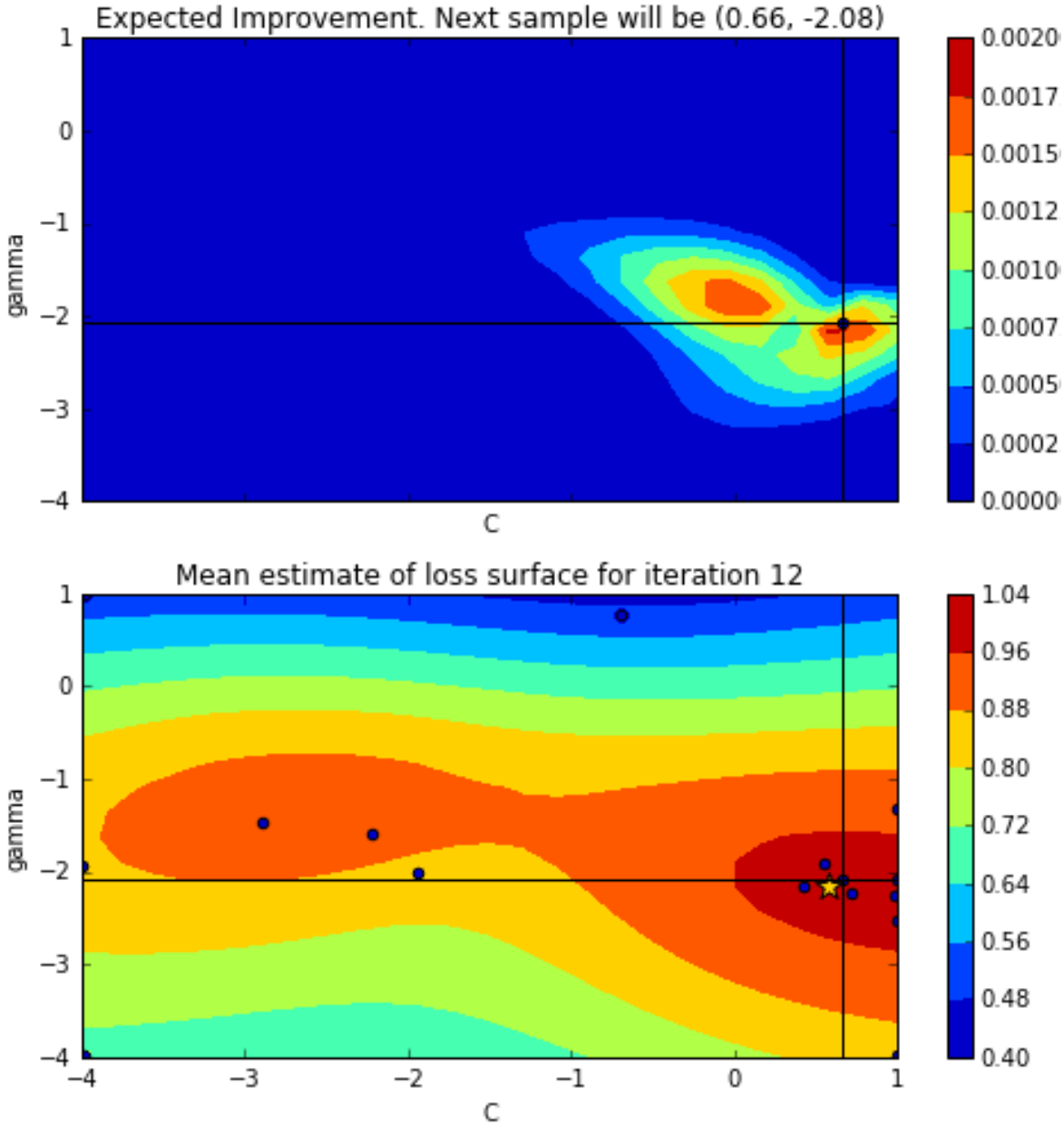
Iterations


Expected Improvement. Next sample will be (0.55, -1.91)
gamma
C
0.0032
0.0028
0.0024
0.0020
0.0016
0.0012
0.0008
0.0004
0.0000
Mean estimate of loss surface for iteration 10
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

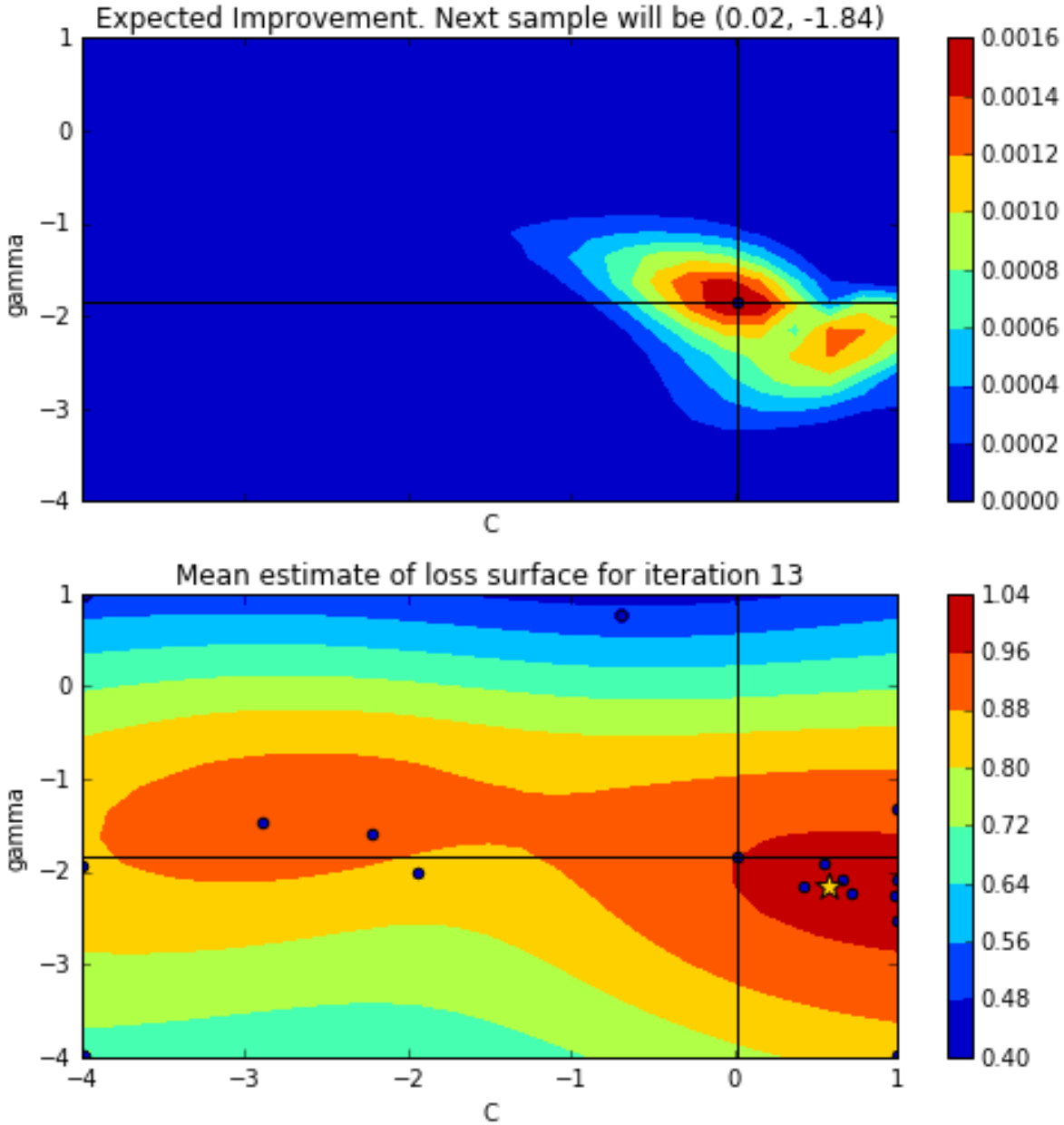
Iterations


Expected Improvement. Next sample will be (0.99, -2.27)
gamma
C
0.0020
0.0017
0.0015
0.0012
0.0010
0.0007
0.0005
0.0002
0.0000
Mean estimate of loss surface for iteration 11
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

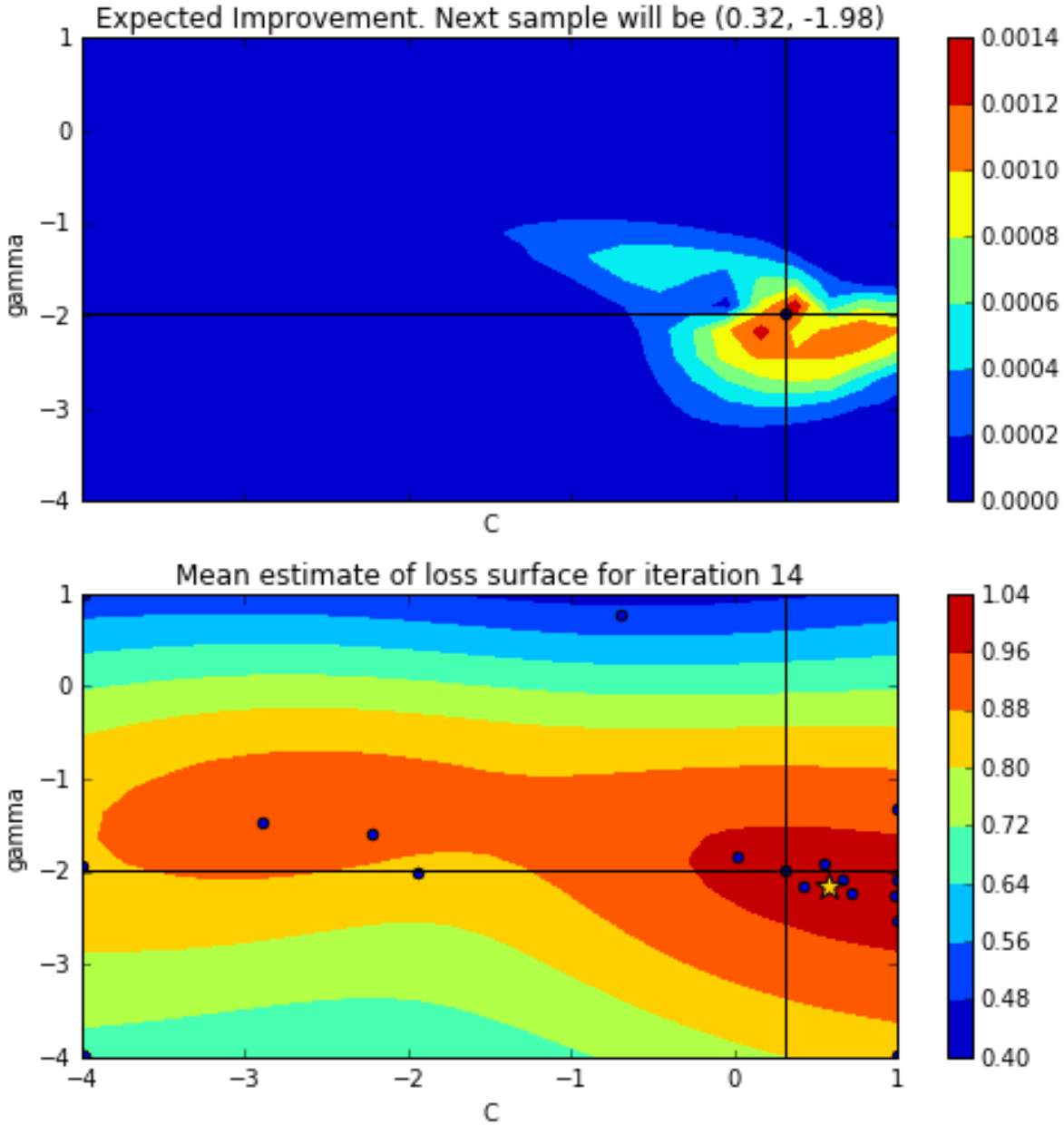
Iterations


Expected Improvement. Next sample will be (0.66, -2.08)
gamma
C
0.0020
0.0017
0.0015
0.0012
0.0010
0.0007
0.0005
0.0002
0.0000
Mean estimate of loss surface for iteration 12
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

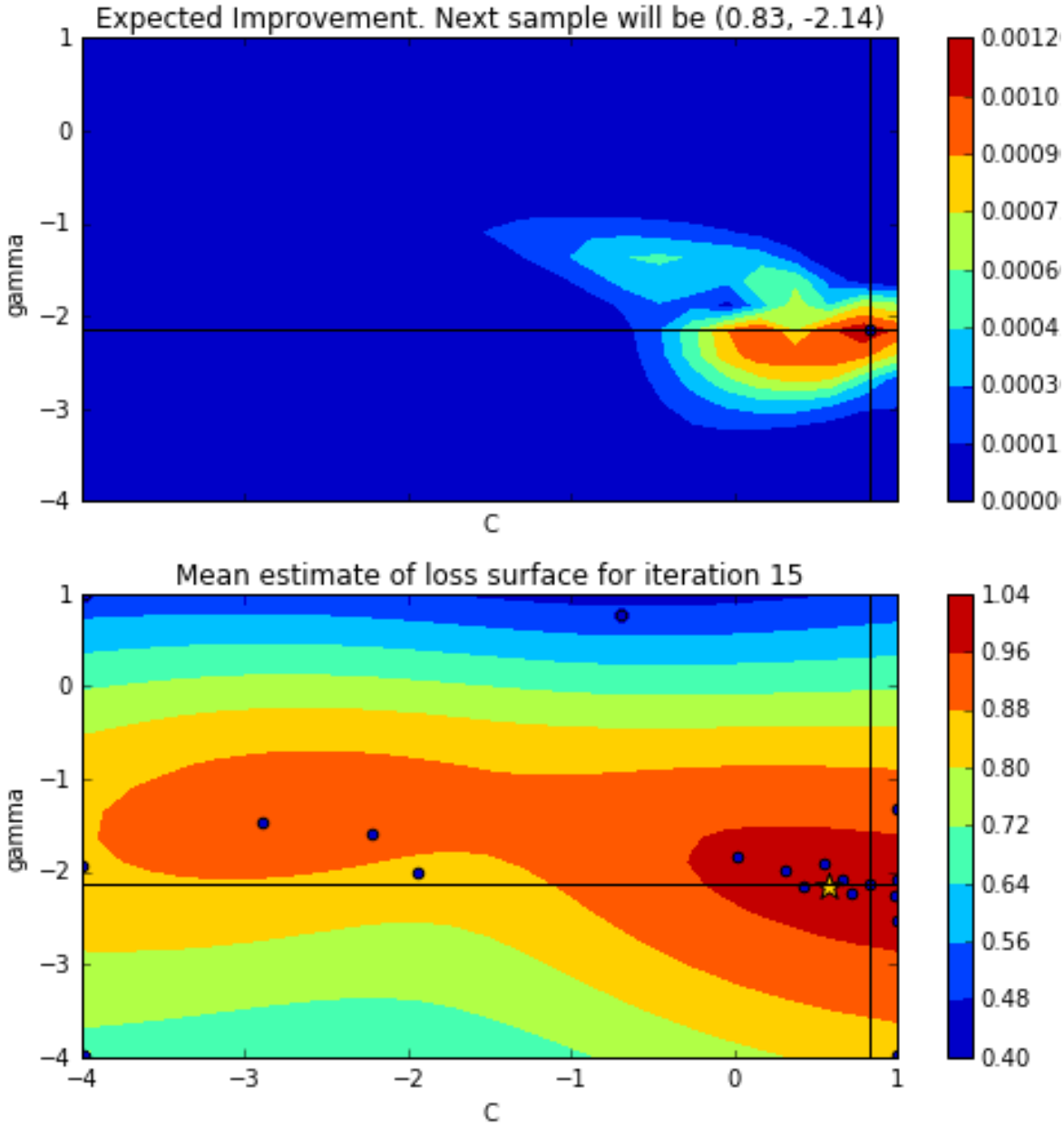
Iterations


Expected Improvement. Next sample will be (0.02, -1.84)
gamma
C
0.0016
0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Mean estimate of loss surface for iteration 13
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

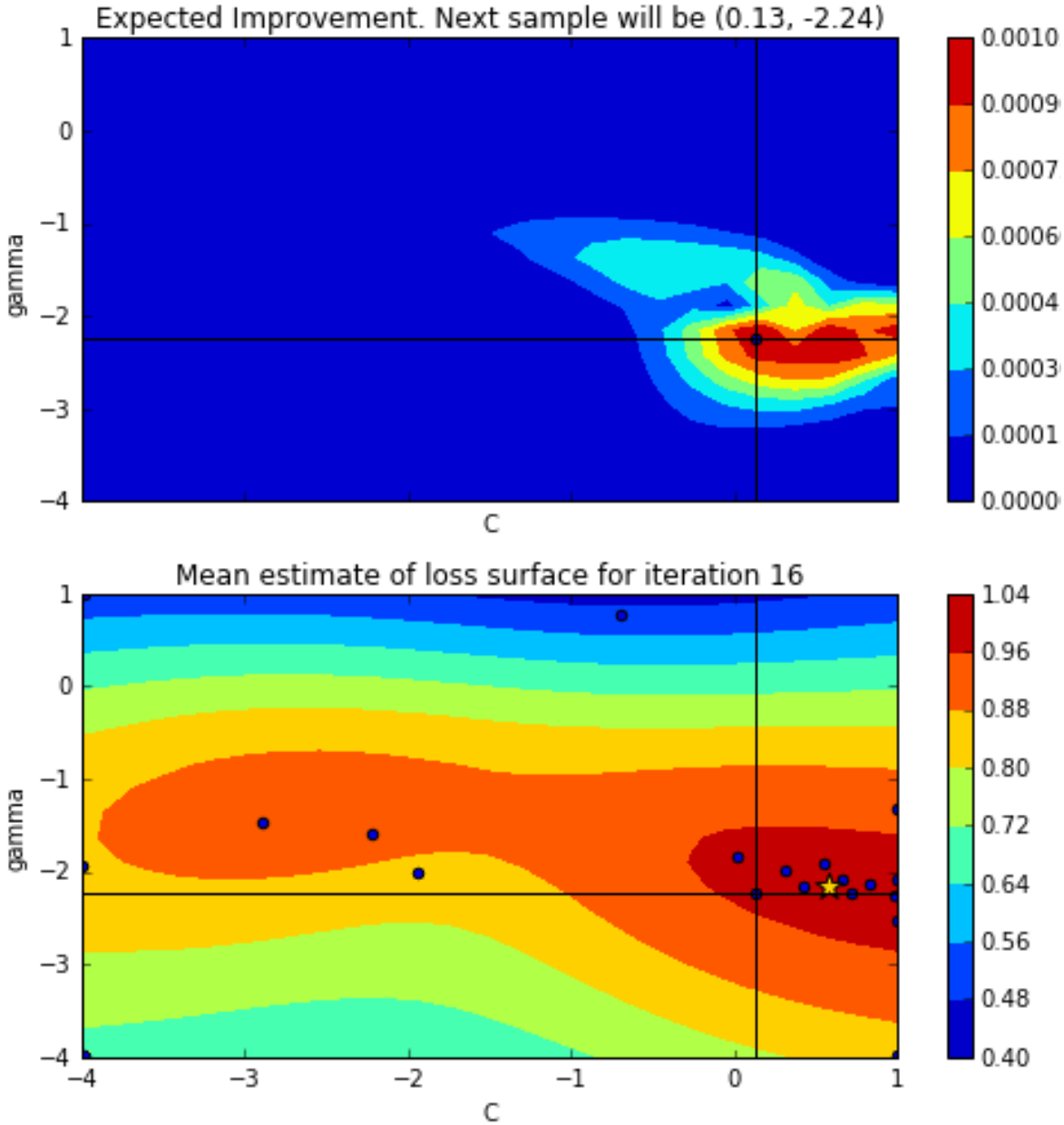
Iterations


Expected Improvement. Next sample will be (0.32, -1.98)
gamma
C
0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Mean estimate of loss surface for iteration 14
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

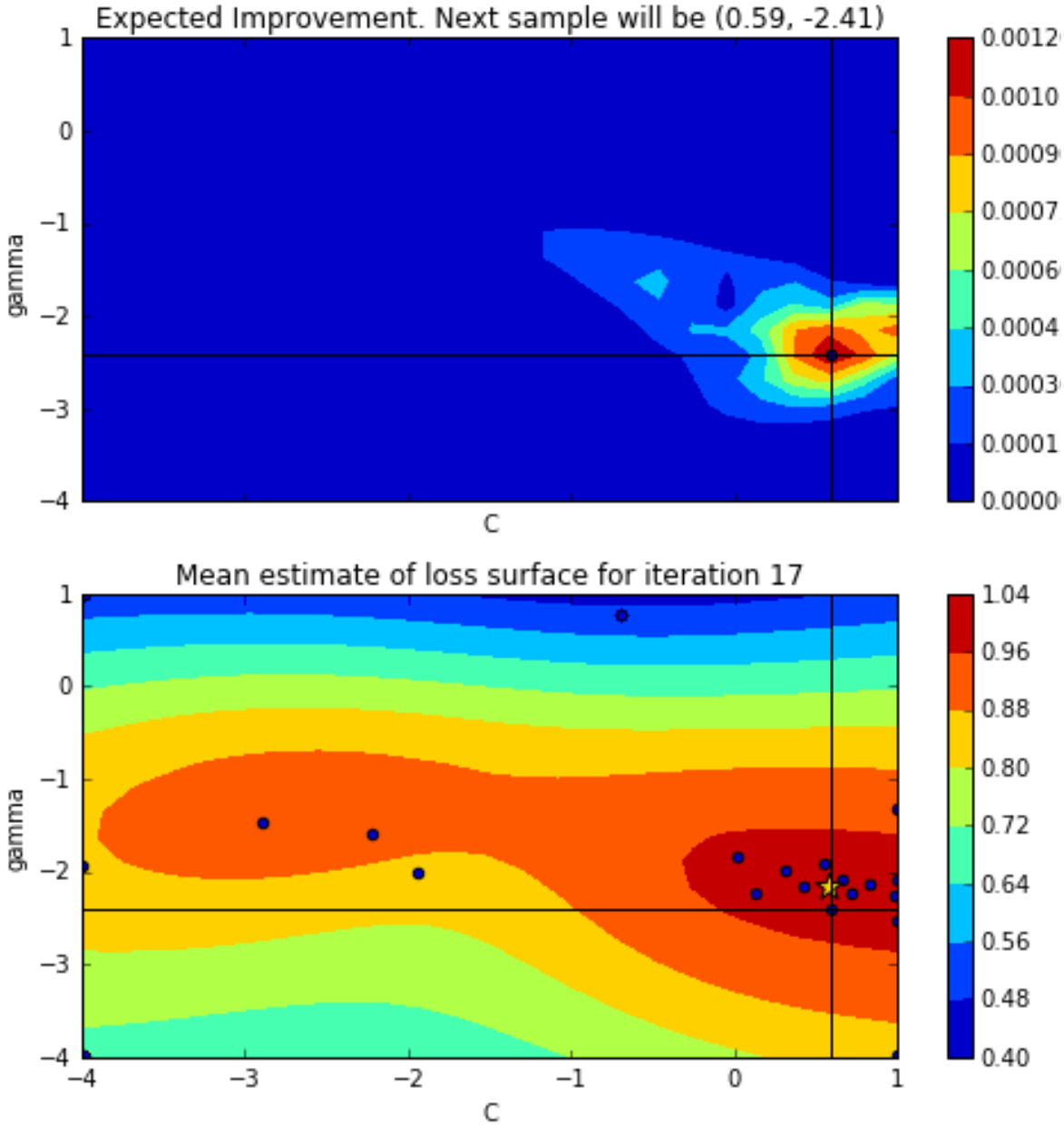
Iterations


Expected Improvement. Next sample will be (0.83, -2.14)
gamma
C
0.0012
0.0010
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 15
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

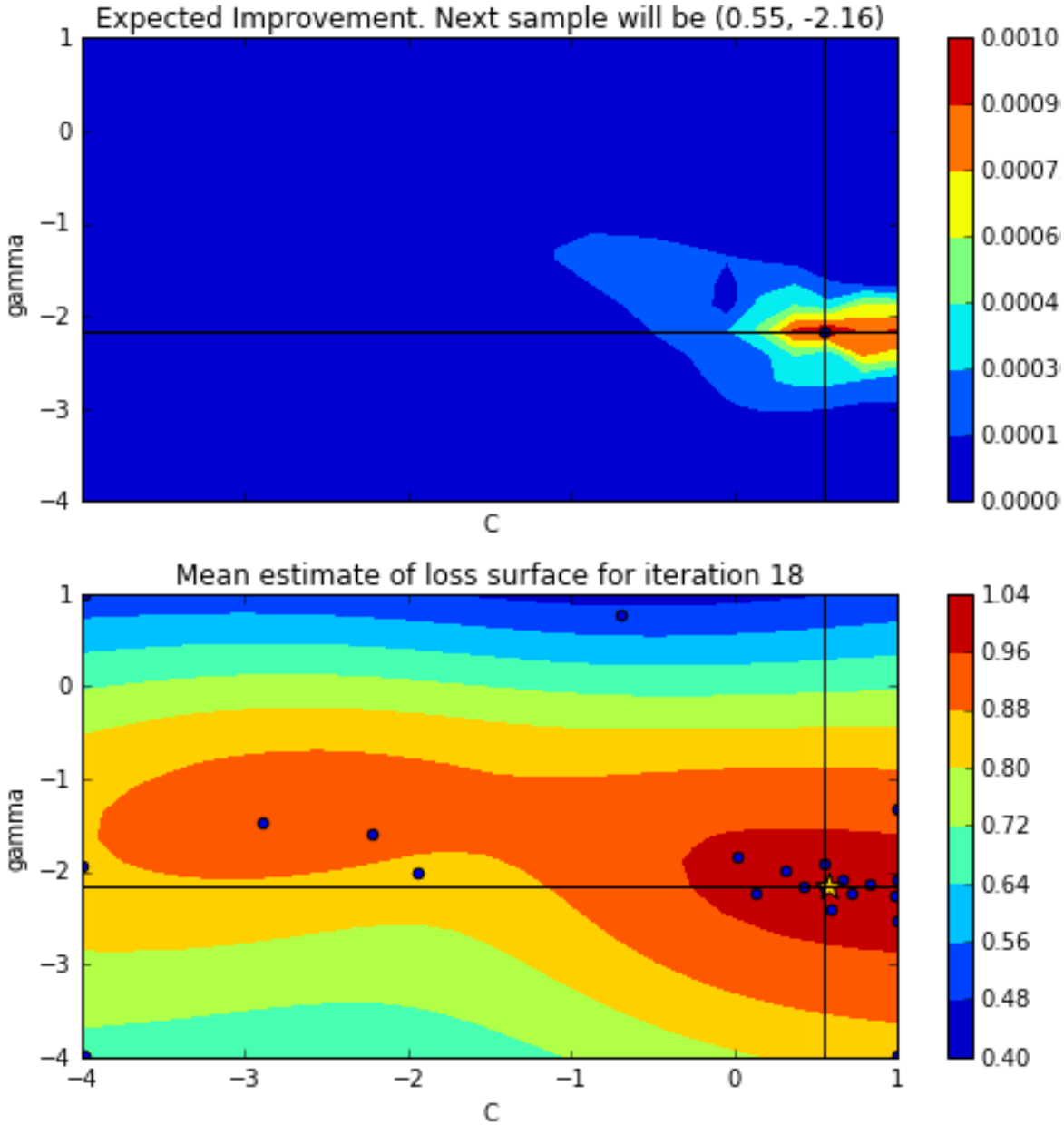
Iterations


Expected Improvement. Next sample will be (0.13, -2.24)
gamma
C
0.0010
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 16
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

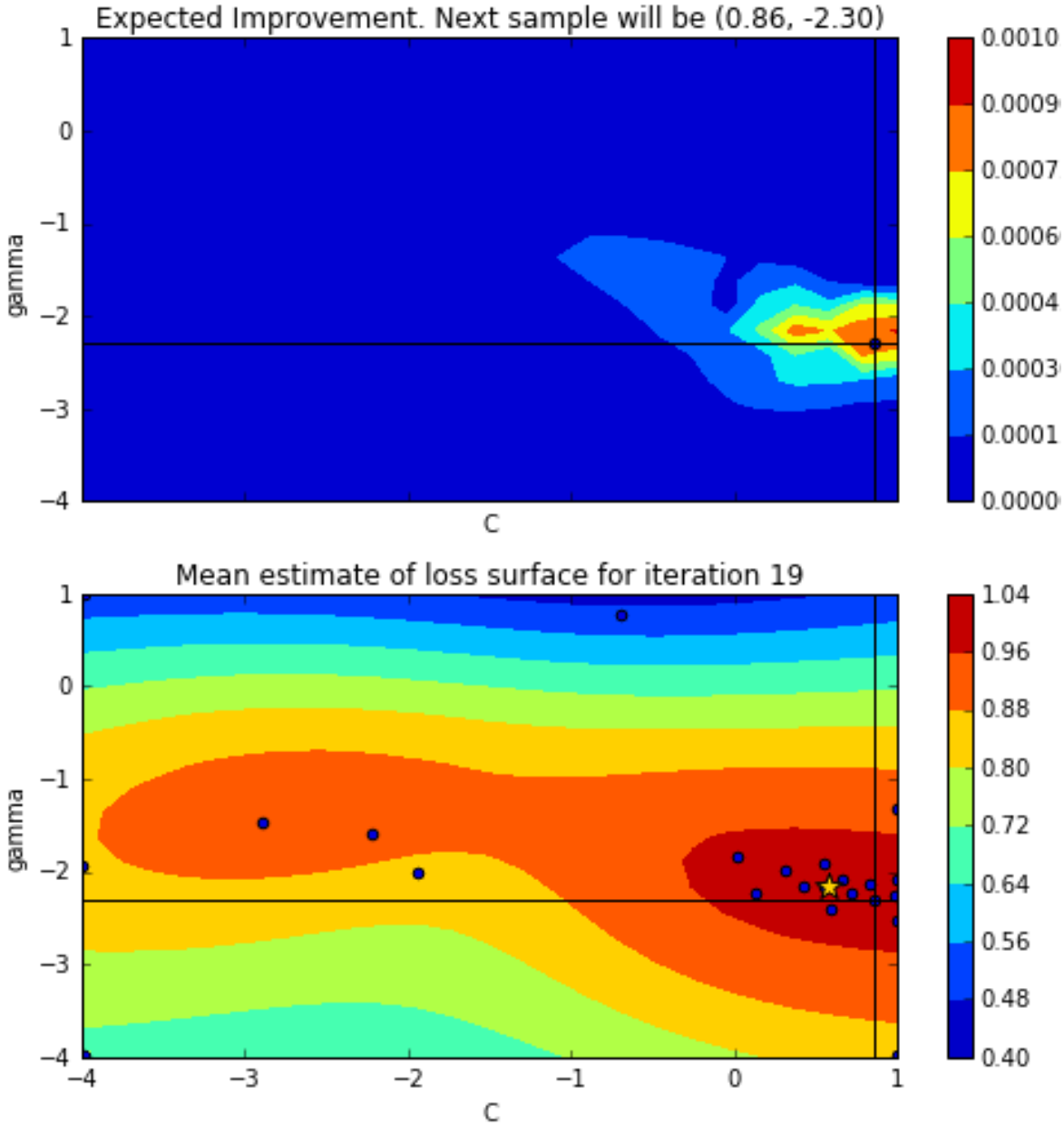
Iterations


Expected Improvement. Next sample will be (0.59, -2.41)
gamma
C
0.0012
0.0010
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 17
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

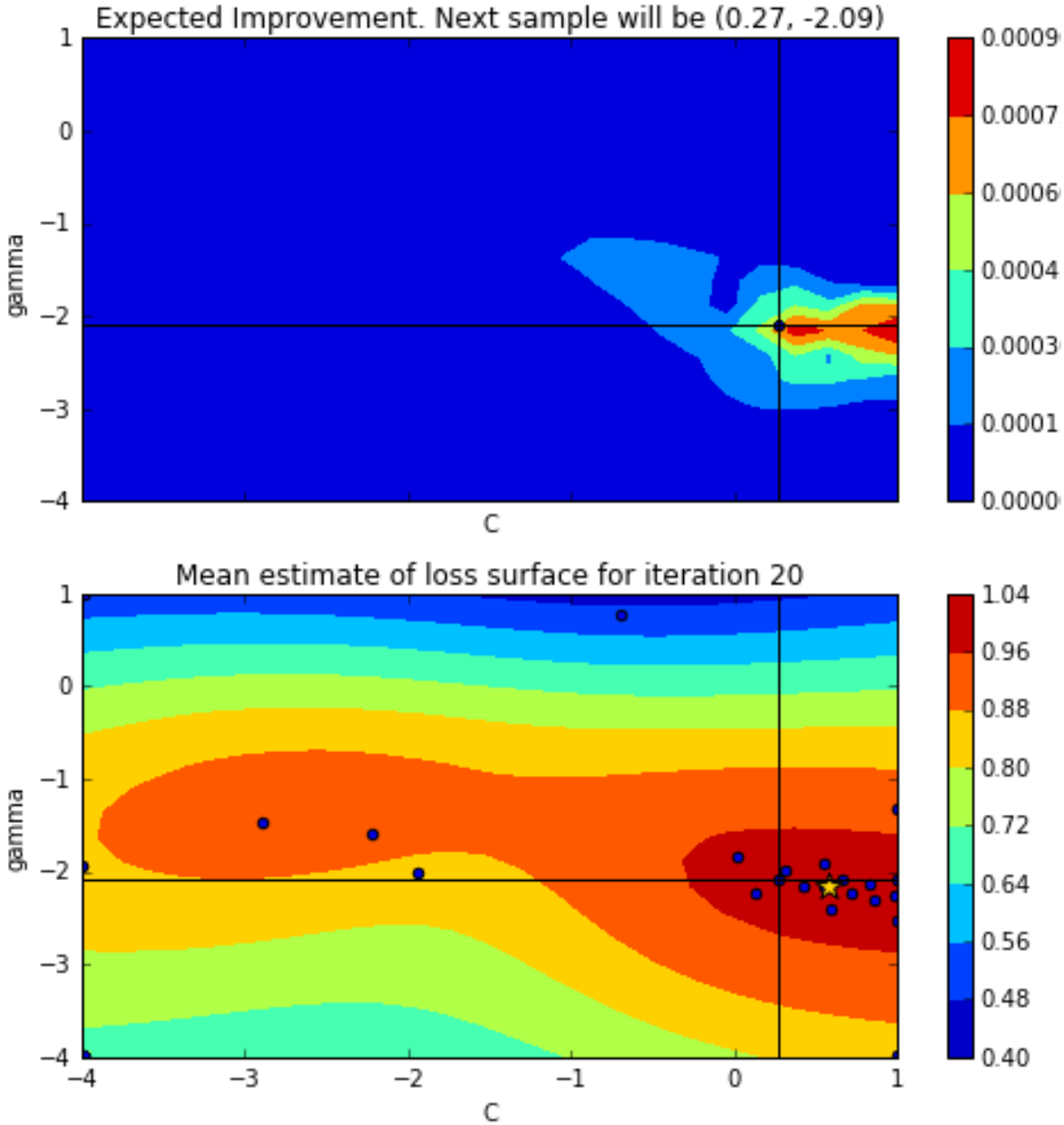
Iterations


Expected Improvement. Next sample will be (0.55, -2.16)
gamma
C
0.0010
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 18
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

Iterations


Expected Improvement. Next sample will be (0.86, -2.30)
gamma
C
0.0010
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 19
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

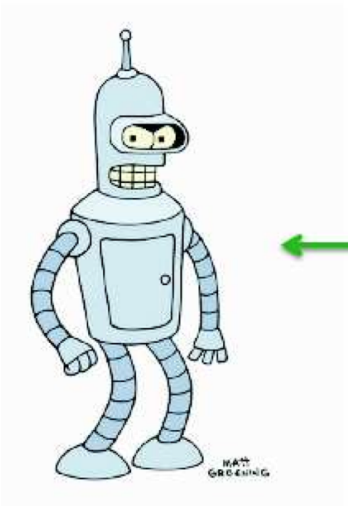
Iterations


Expected Improvement. Next sample will be (0.27, -2.09)
gamma
C
0.0009
0.0007
0.0006
0.0004
0.0003
0.0001
0.0000
Mean estimate of loss surface for iteration 20
gamma
C
1.04
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40

(Passive) Supervised Learning

raw unlabeled data
$x_1, x_2, x_3, \ldots$

random sample



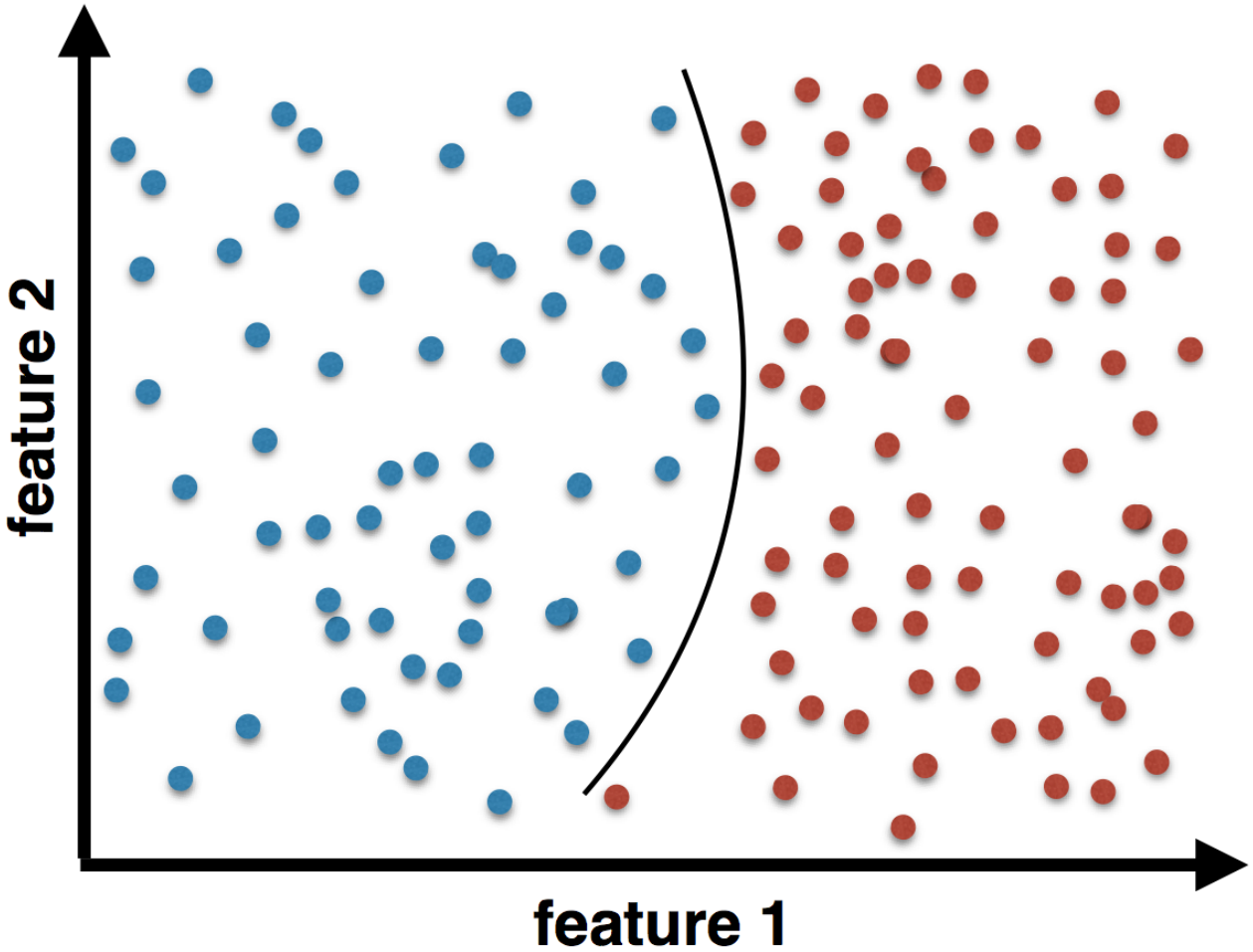
labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

supervised learner
induces a classifier

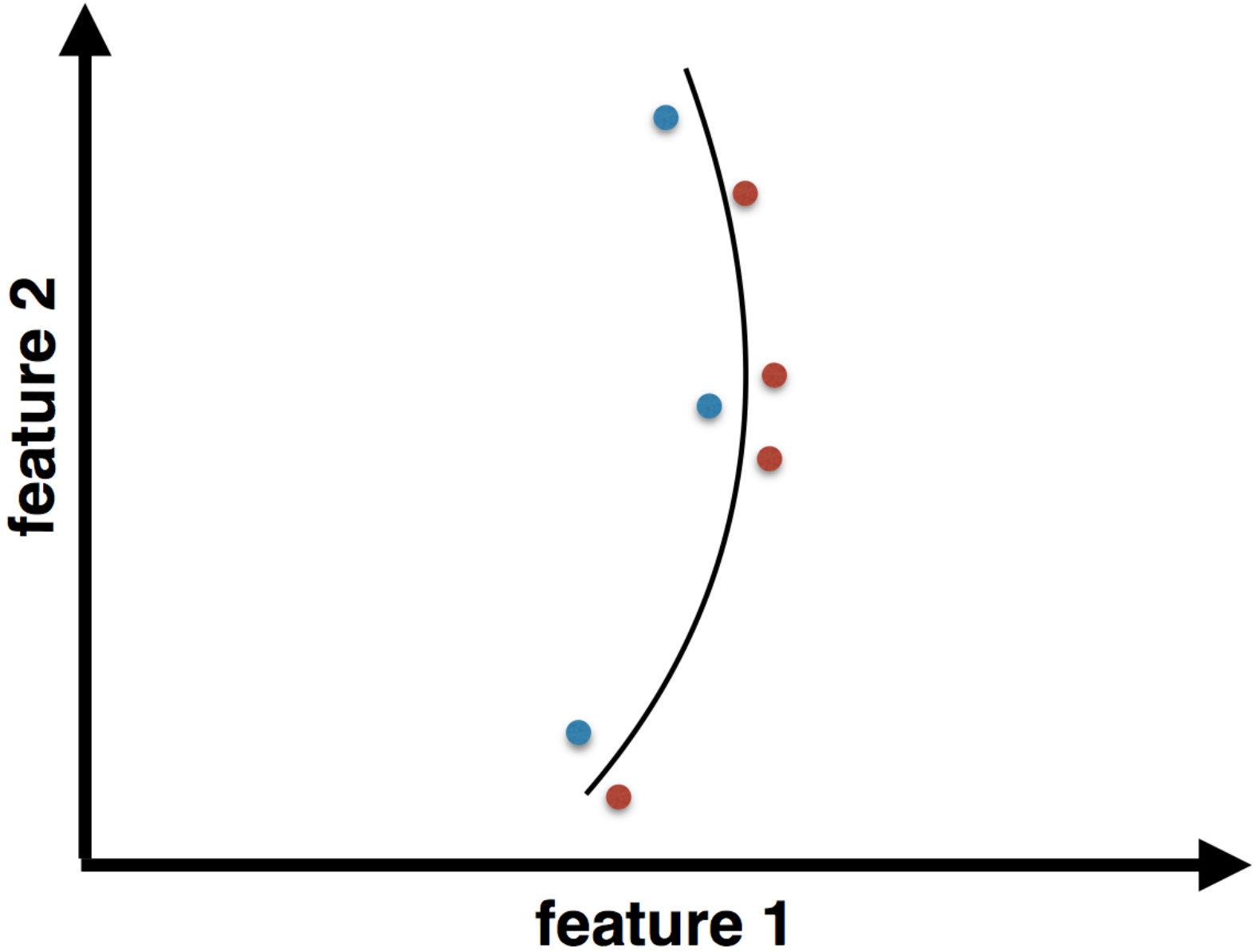
expert / oracle
analyzes experiments
to determine labels

So what's wrong with Supervised Learning

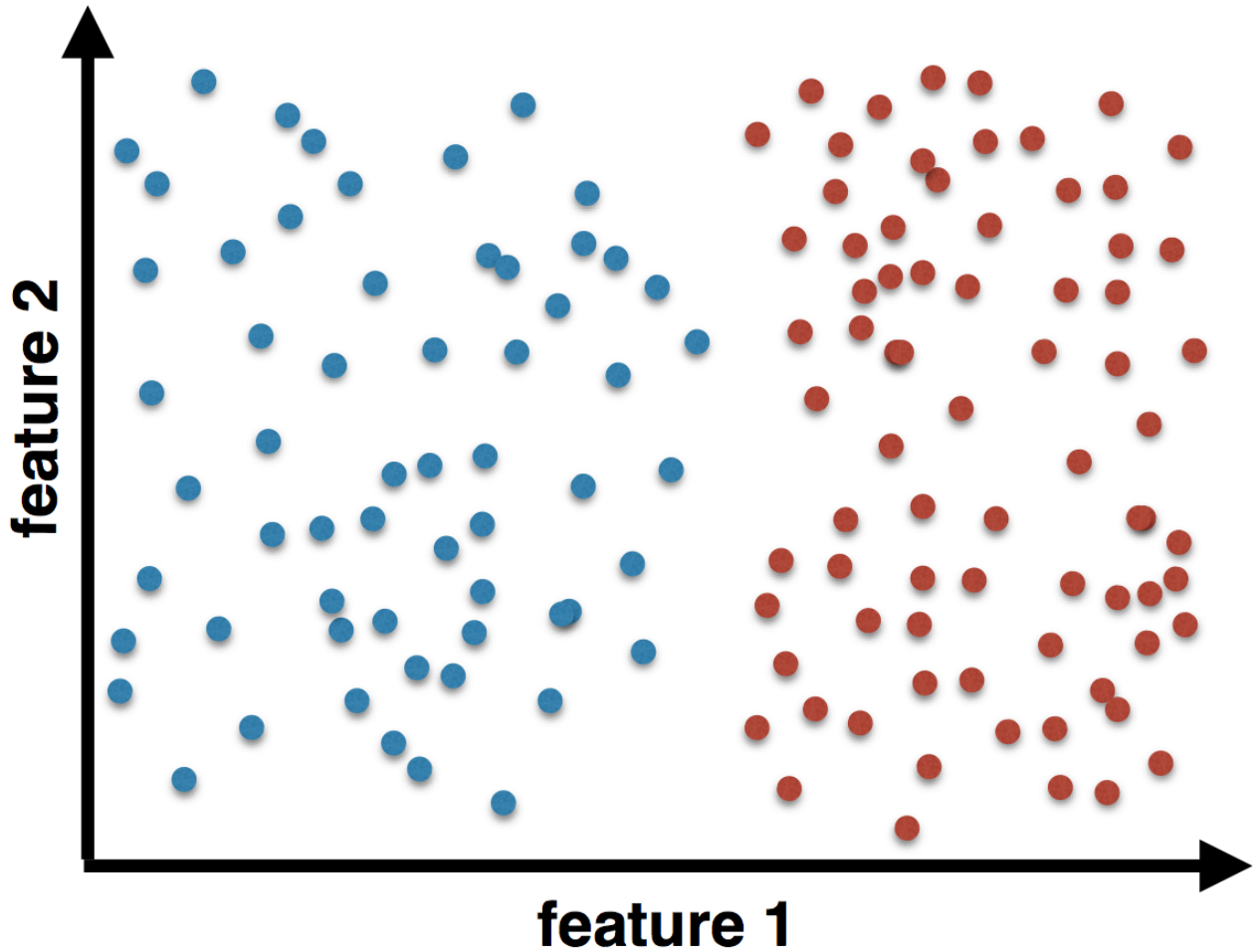


- Traditional approach almost universally adopted

Well, we actually only needed this!

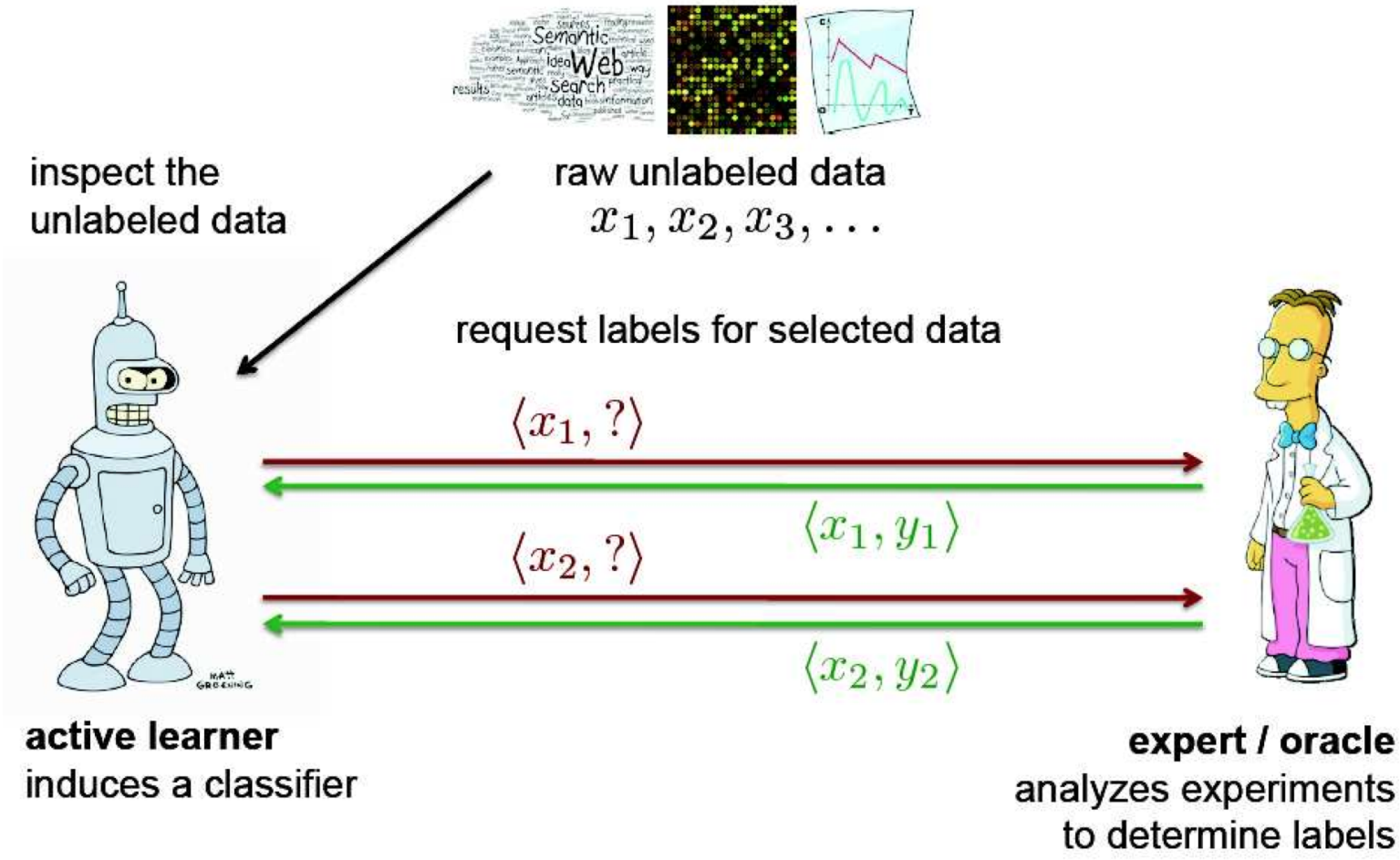


So this was a complete waste of time!



- Random sampling inevitably leads to redundancy

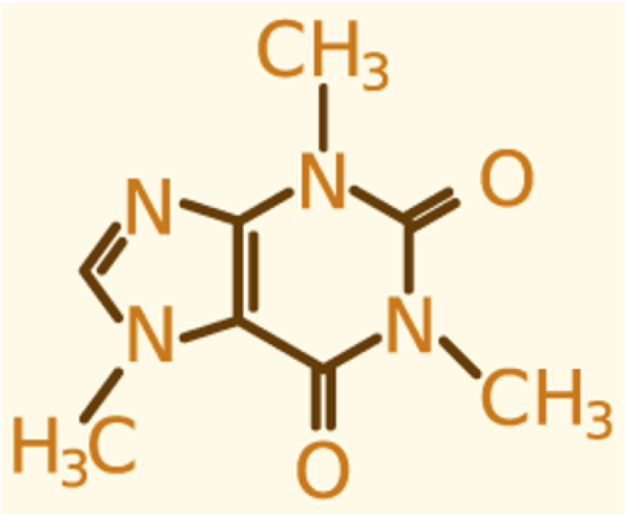
Active Learning



Aim of Active Learning: increase accuracy as much as possible while using a less as possible additional labeled examples

Active Learning example: drug design

- Goal: find compounds which bind to a particular target



Large collections of compounds from:

- vendor catalogs
- corporate collections
- combinatorial chemistry

unlabeled point $\equiv$ description of chemical compound

label $\equiv$ active (binds to target) vs. inactive

getting a label $\equiv$ chemistry experiment

Definition

Adaptive DoE is an approach that allows to iteratively add points to the training set minimizing the error of the model.

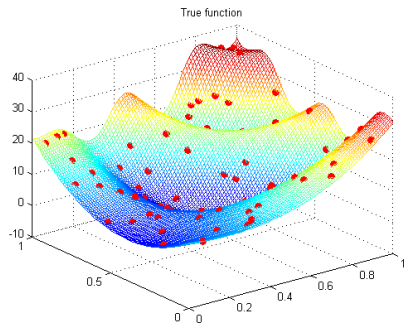


Figure – Unknown function

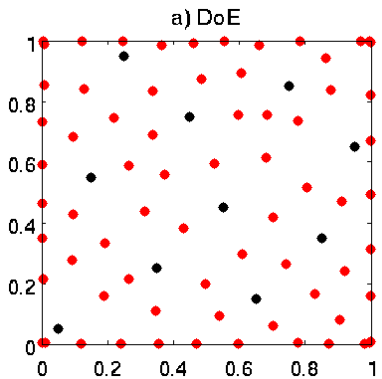


Figure – Initial and added points

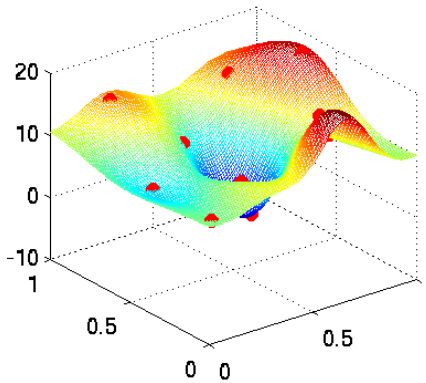


Figure – Initial model

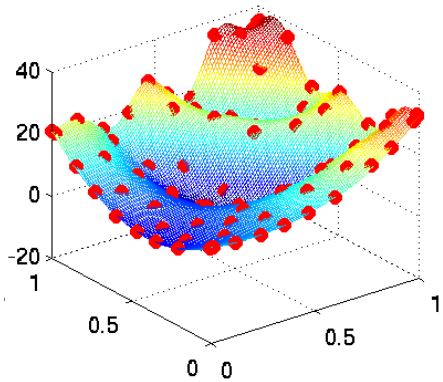


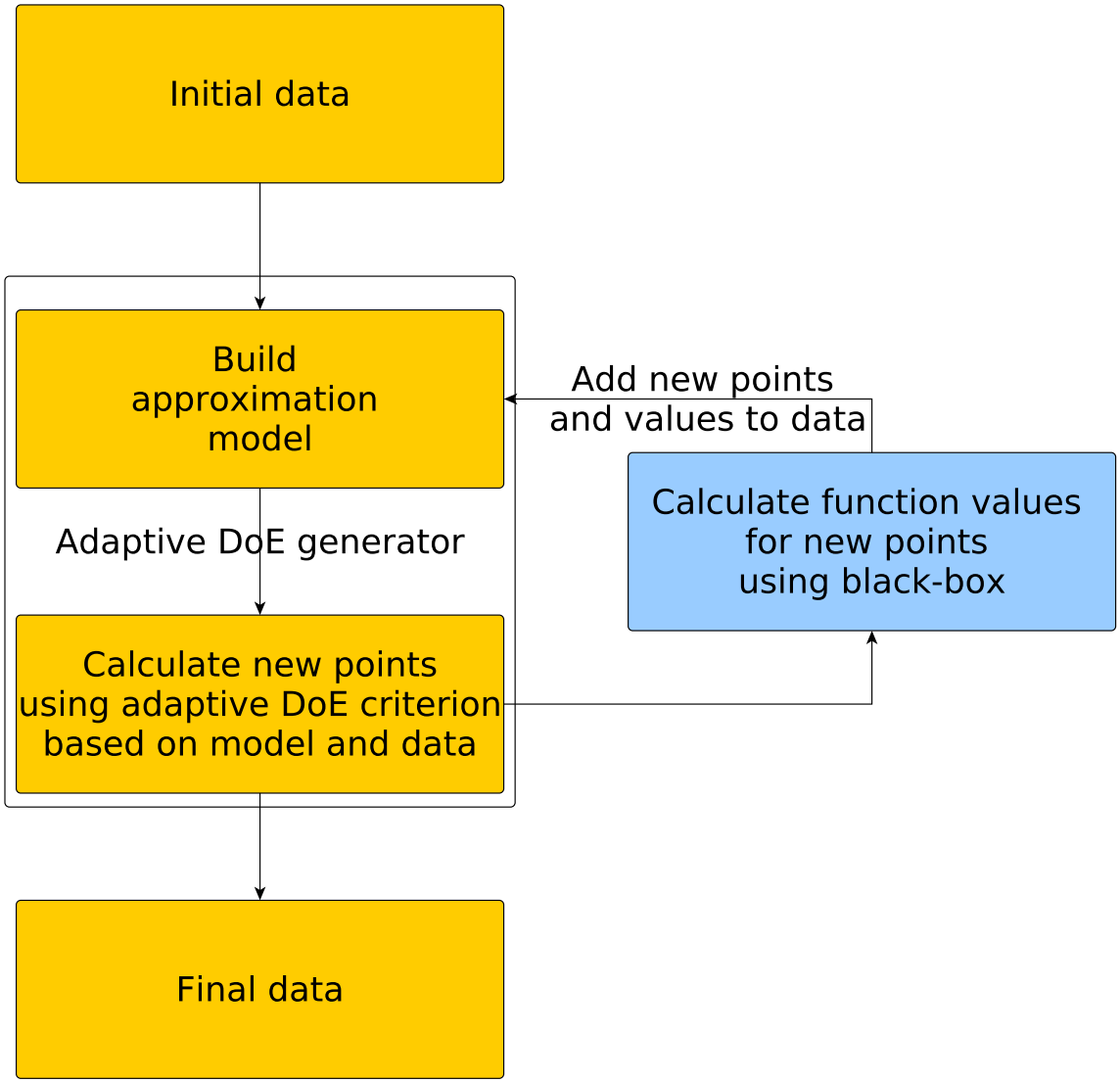
Figure – Final model

Adaptive DoE allows to control the process of modelling by adaptive sampling that improves the quality of the model

Adaptive DoE:

- Generate initial sample using space-filling technique
- Build a model using the obtained training set
- Update the training set by iteratively adding points which improves the model the most

Adaptive DoE



Reasons for such approach:

1. Approximation contains information about the dependency

$$\Downarrow$$

Approximation can give clue on how to select new point in order to increase model quality

2. Adaptive update of training set gives flexibility in sample size

$$\Downarrow$$

If the desired quality is reached generation of DoE can be stopped

Sampling criterion

The procedure of choosing new point can be formulated as maximization of some criterion:

$$\mathbf{x}_{m+1} = \arg\max_{\mathbf{x}\in\mathbb{X}} \mathcal{I}(\mathbf{x}|S_m, \widehat{f}_m, \widehat{\sigma}^2_m),$$

where $\mathcal{I}(\mathbf{x}|S, \widehat{f}, \widehat{\sigma}^2)$ is a criterion of point selection based on

- current training set S,
- current approximation model $\widehat{f}$ and
- current error of approximation $\widehat{\sigma}^2$

Maximum variance criterion

- Maximum variance criterion:

$$\mathcal{I}_{MV}(\mathbf{x}) = \widehat{\sigma}^2(\mathbf{x}|S),$$

  where $\widehat{\sigma}^2(\mathbf{x}|S)$ is an error at point $\mathbf{x}$ of the model trained on $S = (\mathbf{X}, \mathbf{y})$
- As $\widehat{\sigma}^2(\mathbf{x}|S)$ we can use GP-based posterior variance $\sigma_*^2(\mathbf{x})$
- **NB**!: for GP

$$\sigma_*^2(\mathbf{x}) = \sigma_*^2(\mathbf{x}|\mathbf{X}),$$

  i.e. it depends on S explicitly only through $\mathbf{X}$

Advantages

- Easy to calculate
- Takes into account information about current approximation

Disadvantages

- Takes into account only local behavior
- Tends to sample points that are close to the boundary $\mathbb{X}$

Minimization of mean prediction error

- GP-based criterion
- Minimum mean squared error on next iteration:

$$\mathcal{I}_{\rho_2}(\mathbf{x}) = \frac{1}{|\mathbb{X}|}\int_{\mathbb{X}}\big(\sigma_*^2(\mathbf{v}|\mathbf{X}) - \sigma_*^2(\mathbf{v}|\mathbf{X}\cup\mathbf{x})\big)\mathrm{d}\mathbf{v},$$

  where
  - $\sigma_*^2(\mathbf{v}|\mathbf{X})$ is a GP posterior variance at point $\mathbf{v}$ of the model built using $S = (\mathbf{X}, \mathbf{y})$
  - $\sigma_*^2(\mathbf{v}|\mathbf{X}\cup\mathbf{x})$ is a GP posterior variance for the input sample $\mathbf{X}^{ext} = \mathbf{X}\cup\mathbf{x}$

- GP-based criterion
- Integrated MSE Gain-Maximum Variance:

$$\mathcal{I}_{IGMV}(\mathbf{x}) = \mathcal{I}_{\rho_2}(\mathbf{x}) \cdot \mathcal{I}_{MV}(\mathbf{x})$$

Advantages

- Takes into account information about model behavior in all the design space

Disadvantages

- Relatively hard to compute

The criterion can be rewritten in the following form:

$$\mathcal{I}_{\rho_2}(\mathbf{x}) = \frac{1}{|\mathbb{X}|}\int_{\mathbb{X}} \frac{k_*^2(\mathbf{x},\mathbf{v})}{\sigma_*^2(\mathbf{x}|\mathbf{X})}\mathrm{d}\mathbf{v},$$

where

$$k_*(\mathbf{x},\mathbf{v}) = K(\mathbf{x},\mathbf{v}) - \mathbf{k}(\mathbf{x})^\top[\mathbf{K}+\sigma^2\mathrm{I}]^{-1}\mathbf{k}(\mathbf{v})$$

is a posterior covariance of GP between values $f(\mathbf{x})$ and $f(\mathbf{v})$

Criterion can be unstable if $\sigma_*^2(\mathbf{x}|\mathbf{X})$ is small

$$\Downarrow$$

It leads to noisy values and multiple non-robust minima

We can solve the problem by combining two criteria: $\mathcal{I}_{\rho_2}(\mathbf{x})$ and $\mathcal{I}_{MV}(\mathbf{x})$

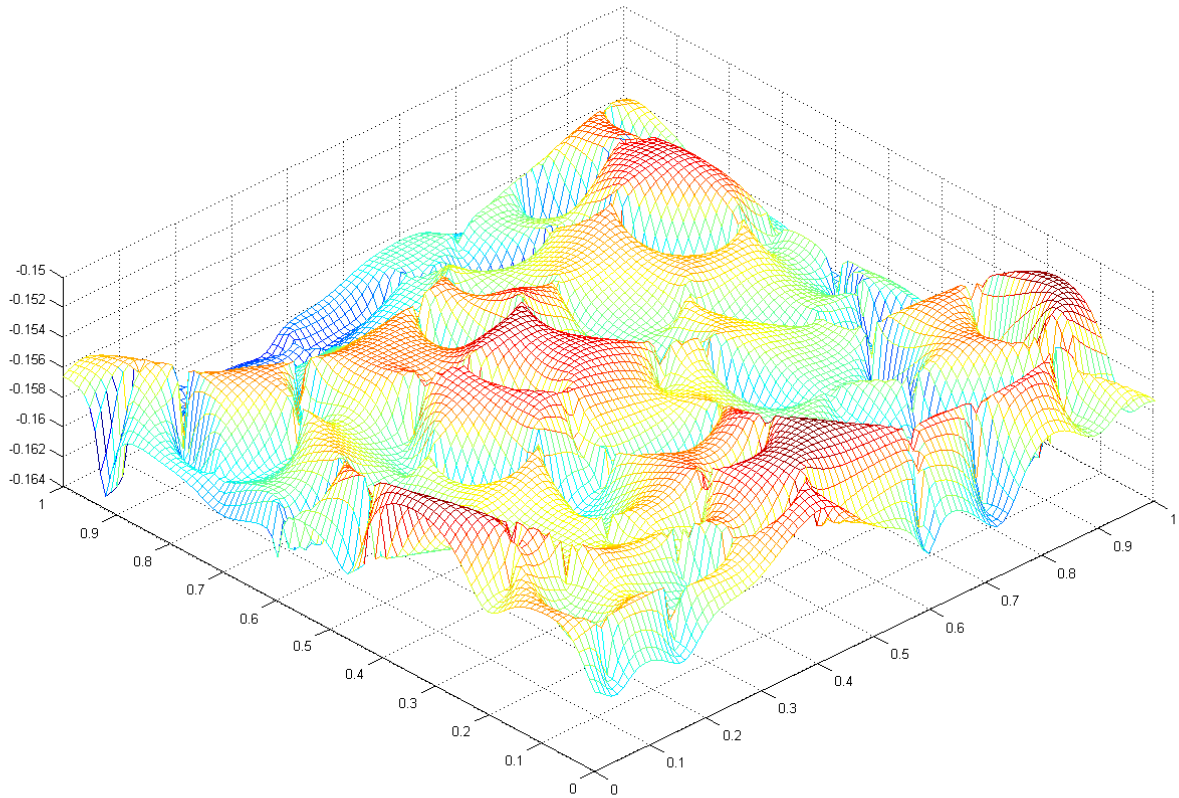
$$\mathcal{I}_{IGMV}(\mathbf{x}) = \mathcal{I}_{\rho_2}(\mathbf{x}) \cdot \mathcal{I}_{MV}(\mathbf{x}) = \frac{1}{|\mathbb{X}|} \int_{\mathbb{X}} k_*^2(\mathbf{x}, \mathbf{v}) \mathrm{d}\mathbf{v}$$

Advantages

- No issues with denominator
- Still takes into account all the design space

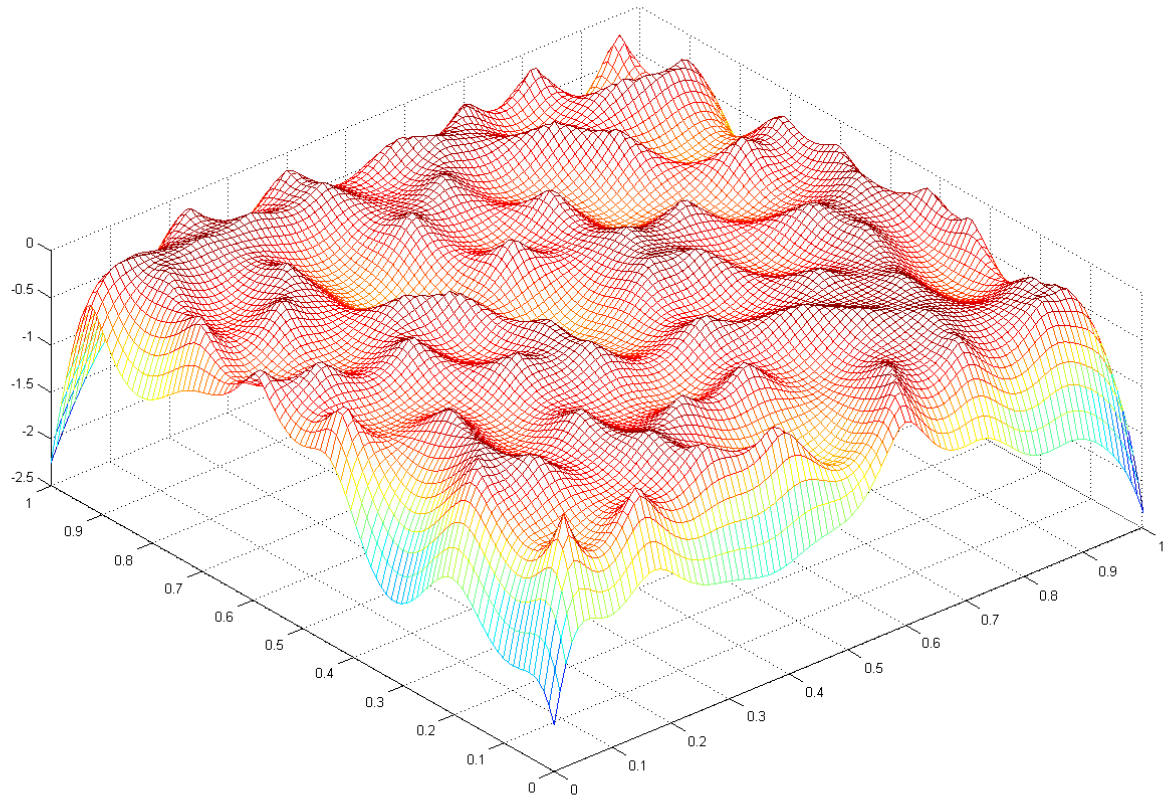
$\mathcal{I}_{\rho 2}$ surface

- High modality
- Narrow and non-robust local minima



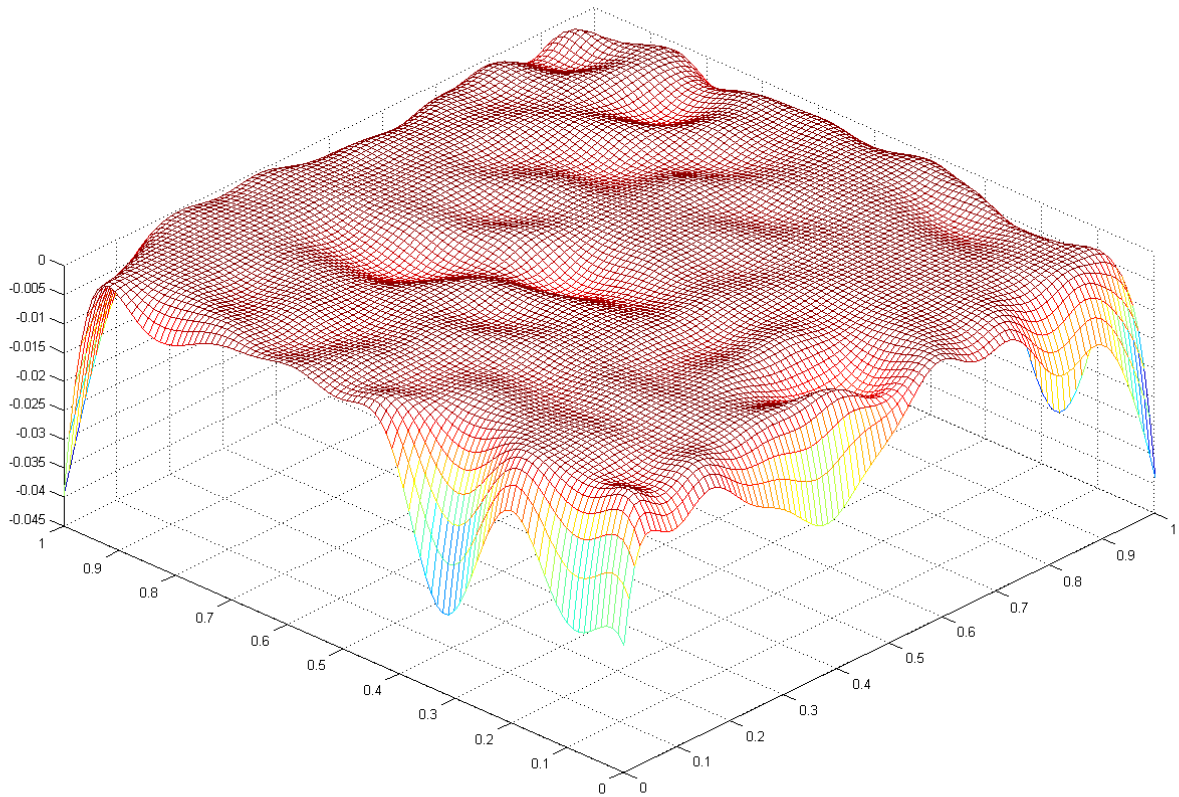
Surface of MaxVar criterion

- High modality
- More regular behavior



Surface of ImseGain-MaxVar

- Less local maxima
- Regular behavior



MaxMin criterion

$$\mathcal{I}_{MM}(\mathbf{x}) = \min_{\mathbf{v}\in\mathbb{X}} d^2(\mathbf{v}, \mathbf{x}),$$

where $d(\mathbf{v}, \mathbf{x})$ is a Euclidean distance between $\mathbf{v}$ и $\mathbf{x}$

Advantages

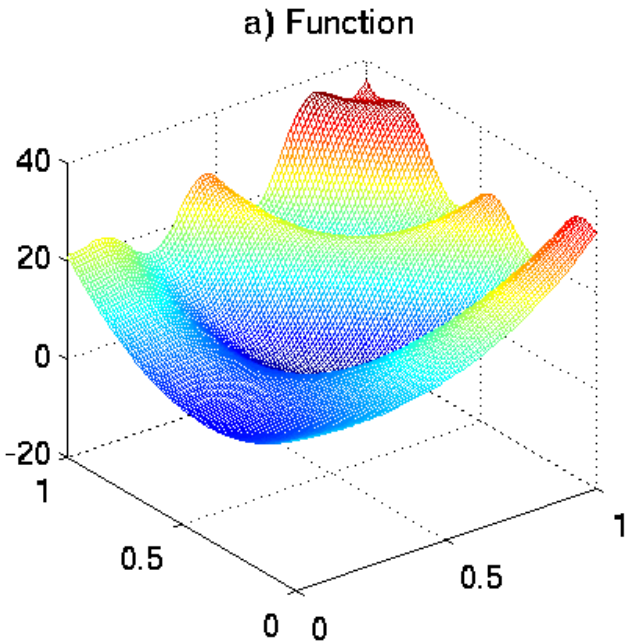
- Very fast computation
- Doesn't depend on the modely type
- Samples point uniformly in the design space

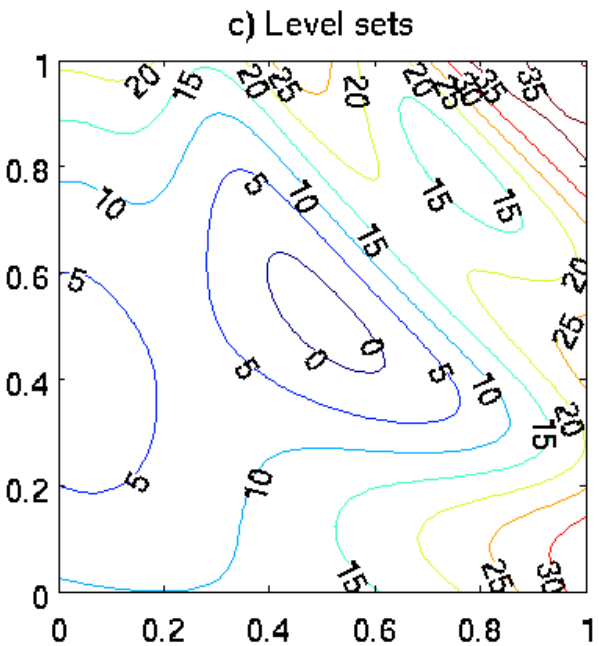
Disadvantages

- Doesn't take into account the model

Example: Toy function

$$y = 2 + 0.25(x_2 - 5x_1^2)^2 + (1 - 5x_1)^2 + 2(2 - 5x_2)^2 + 7\sin(2.5x_1)$$





Example: initial approximation

Initial experimental design contains 10 points
Let us build GP based approximation

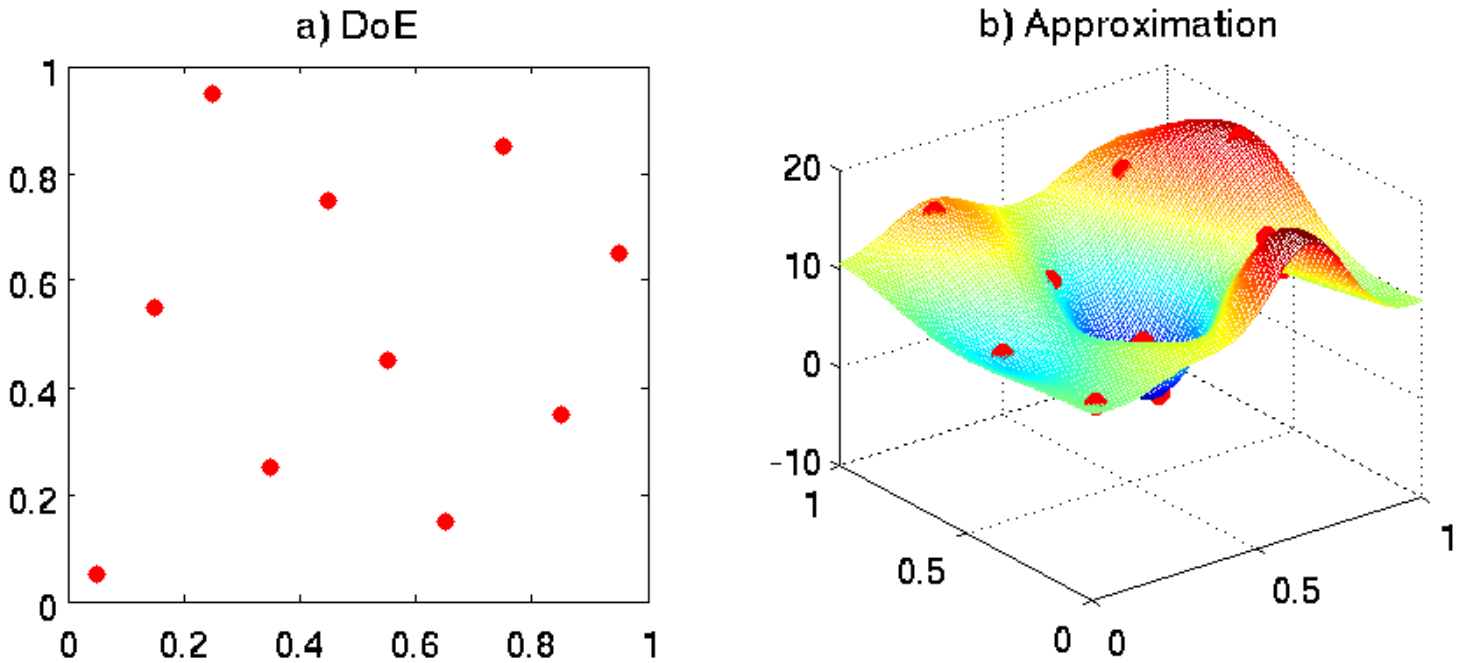


Figure – Initial training set and approximation

70 points were added using MaxVar criterion

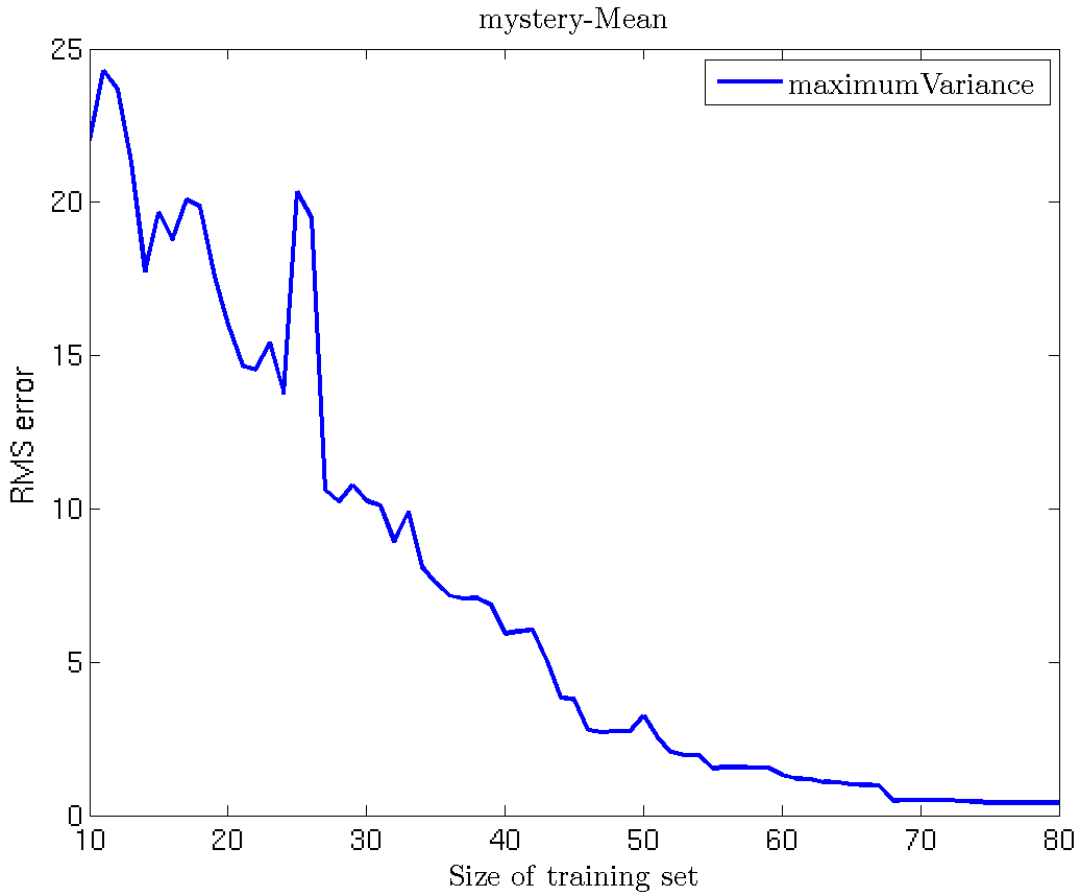


Figure – Approximation error vs size of the training set

Example: final approximation

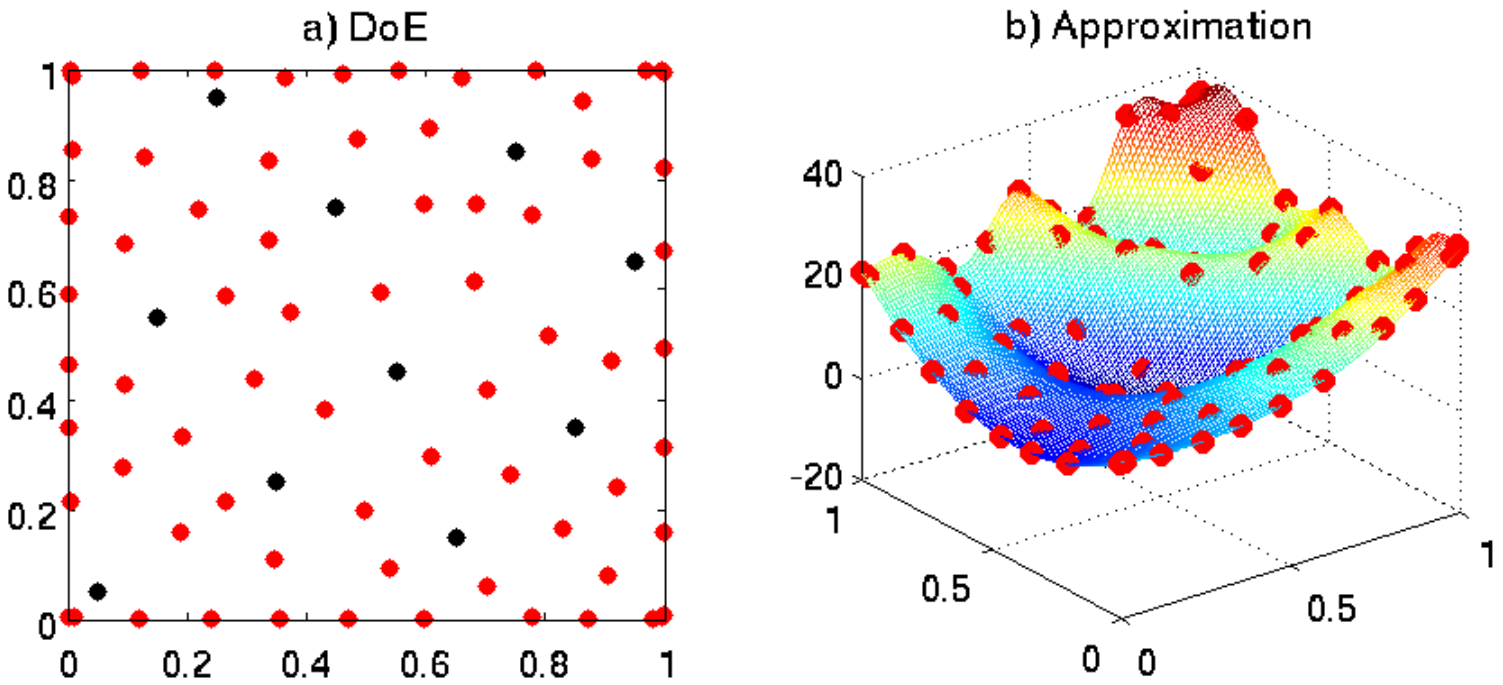


Figure – Final training set and approximation

Example: results

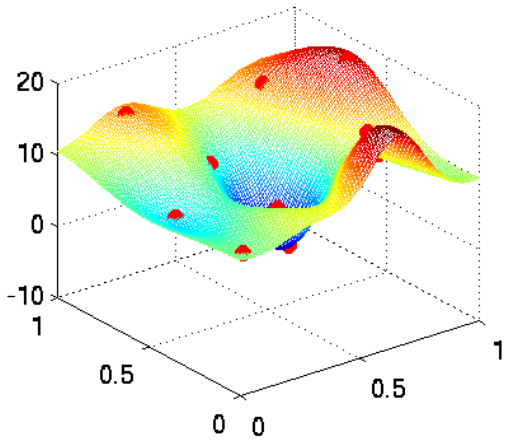


Figure – Initial model

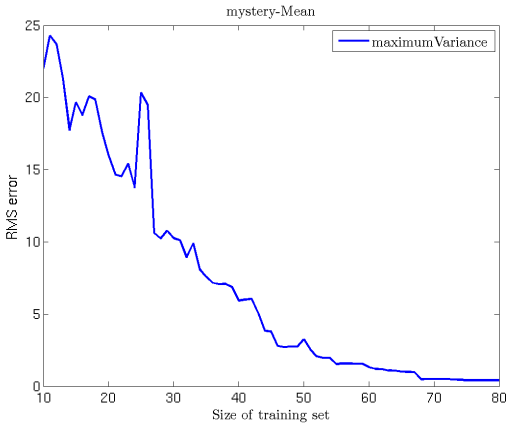


Figure – Error vs training set size

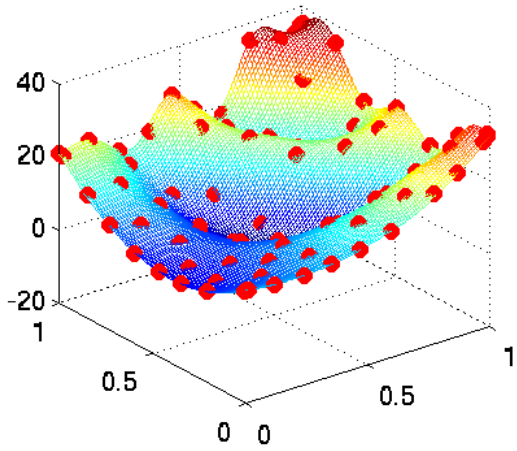


Figure – Final model

Labeling examples

- Example 1: Netflix Challenge
  - Concept: movies Bob would like
  - Instances: $10\,000$ movies on netflix
  - Labeling: Bob watches a movie and reports
- Example 2: Labeling phonemes
  - Concept: words labeled with phonetic alphabet
  - Instances: 1000 hours of talk radio recordings
  - Labeling: Hire linguist to annotate each syllable

TOO TIME CONSUMING/EXPENSIVE!!!

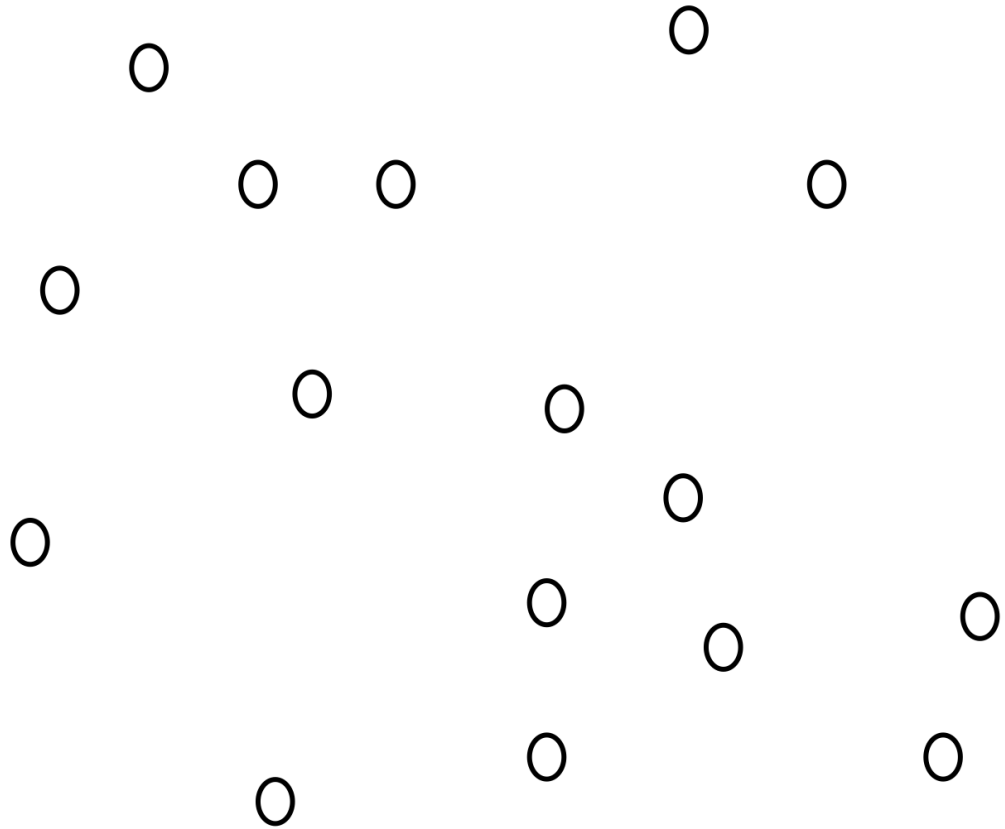
Simple Idea

- If we just pick the RIGHT examples to label, we can learn the concept from only a few labeled examples

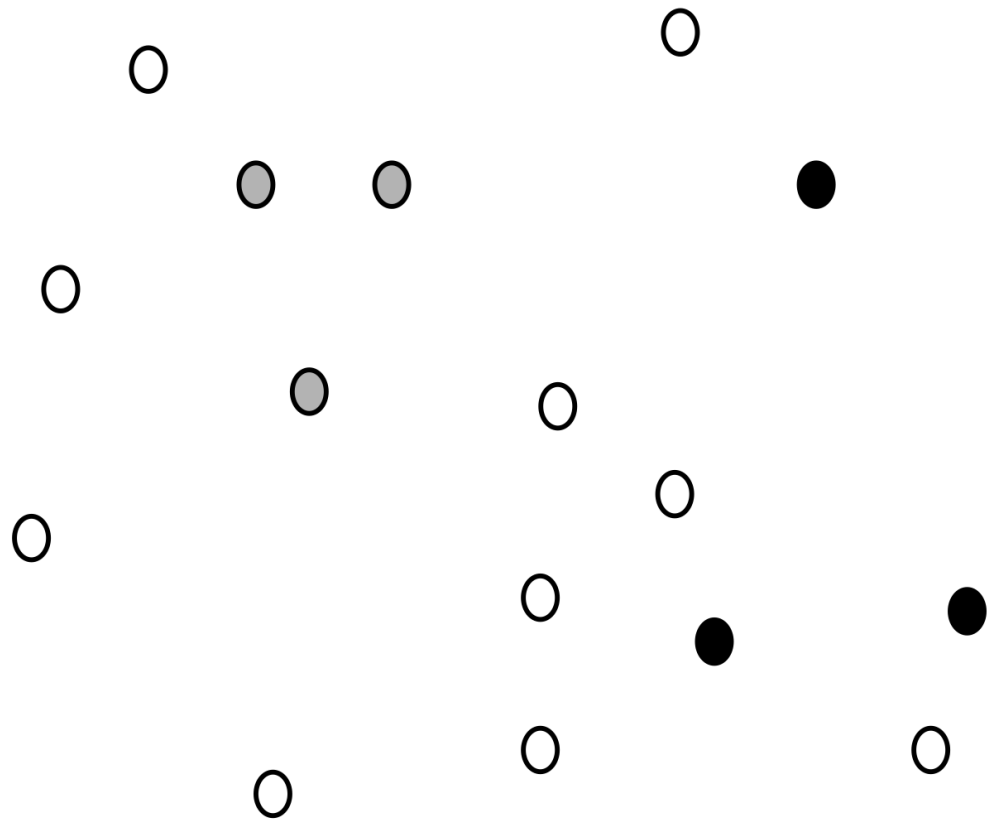
Active Learning Heuristic

- Start with a pool of unlabeled data
- Pick a few points at random and get their labels
- Repeat the following until we have budget left for getting labels
  1. Fit a classifier to the labels seen so far
  2. Pick the BEST unlabeled point to get a label for
     - (closest to the boundary?)
     - (most uncertain?)
     - (most likely to decrease overall uncertainty?)

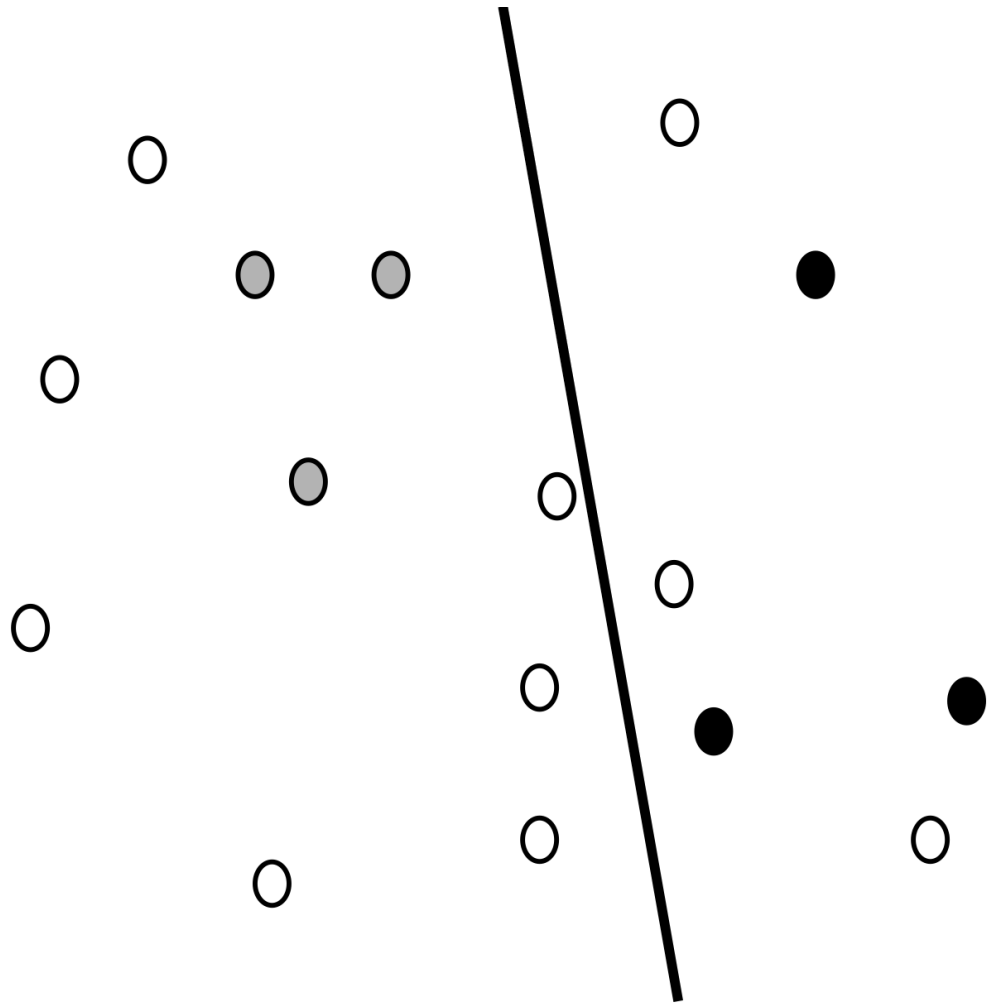
Start: Unlabeled Data



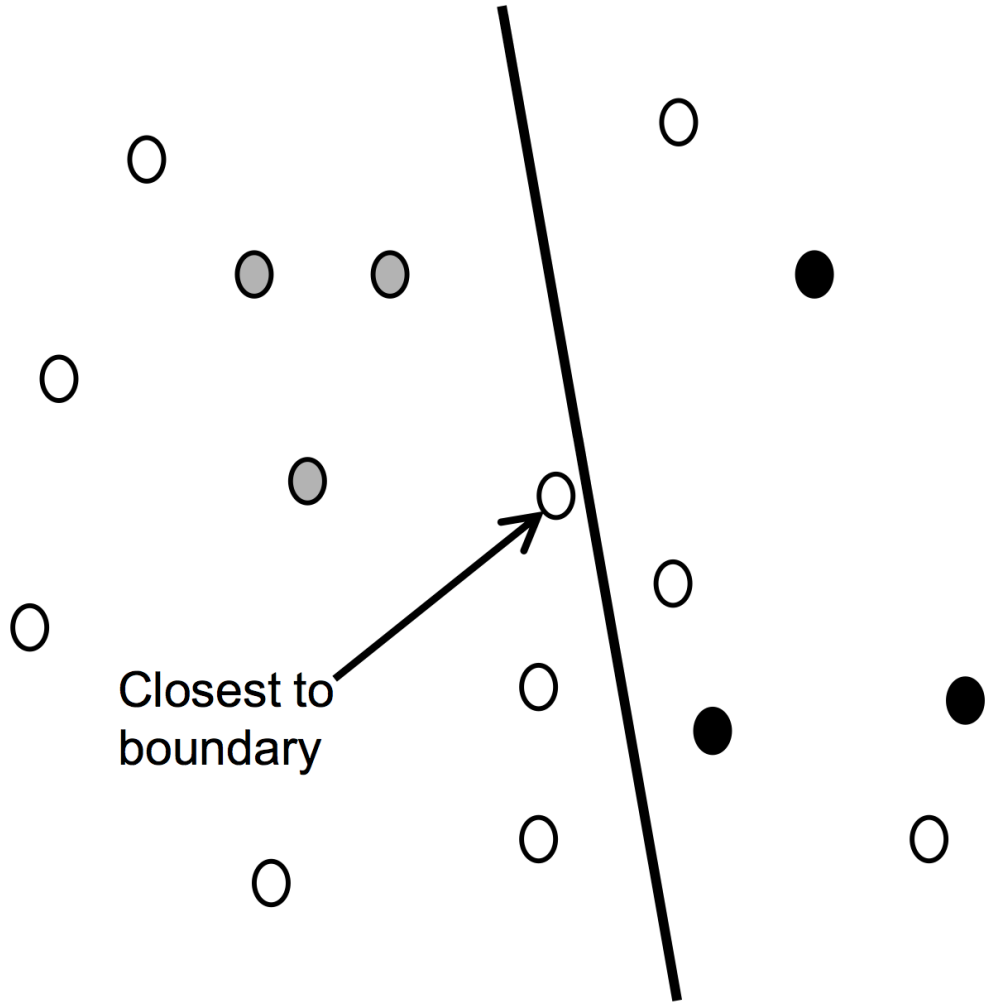
Label a Random Subset



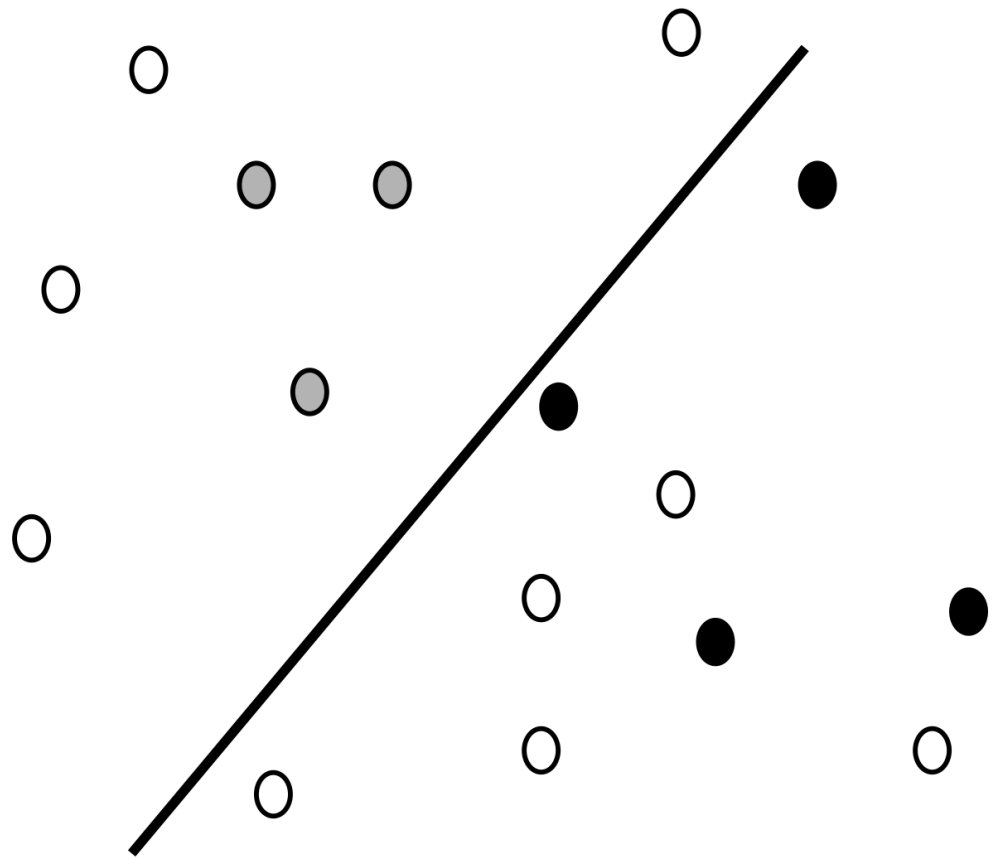
Fit a Classifier to Labeled Data



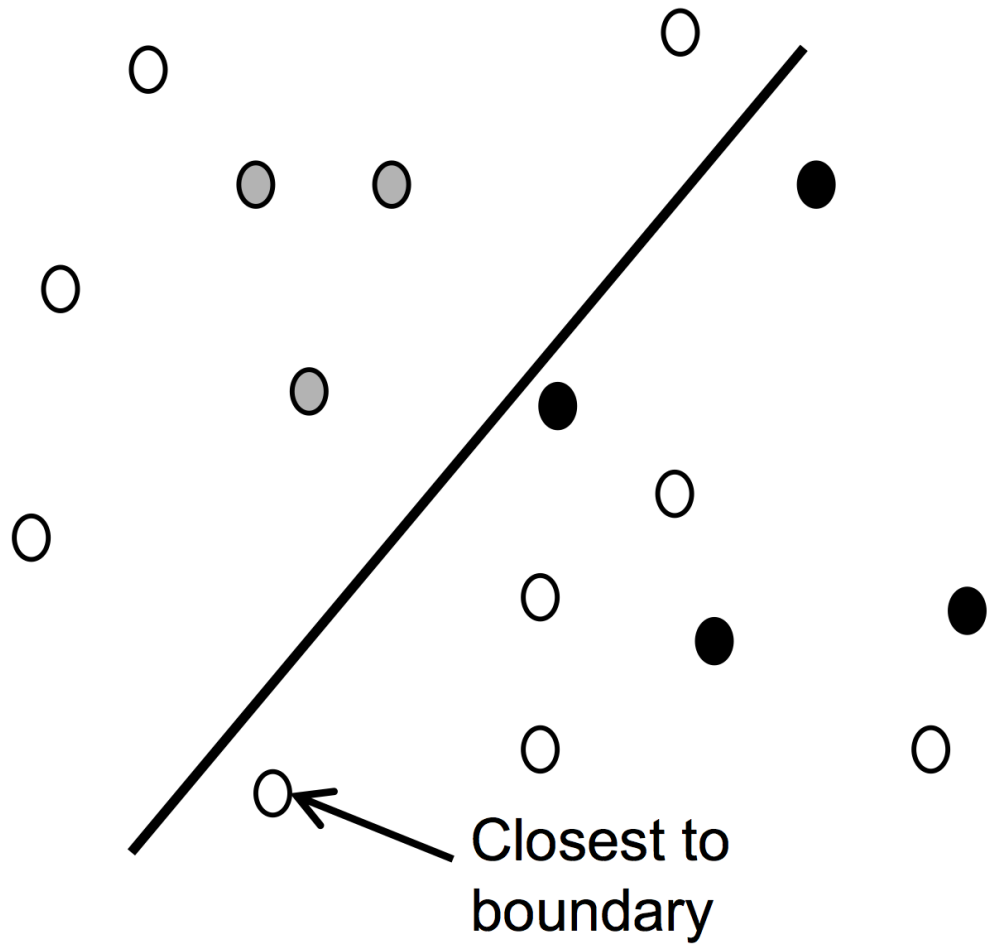
Pick the Best Next Point To Label



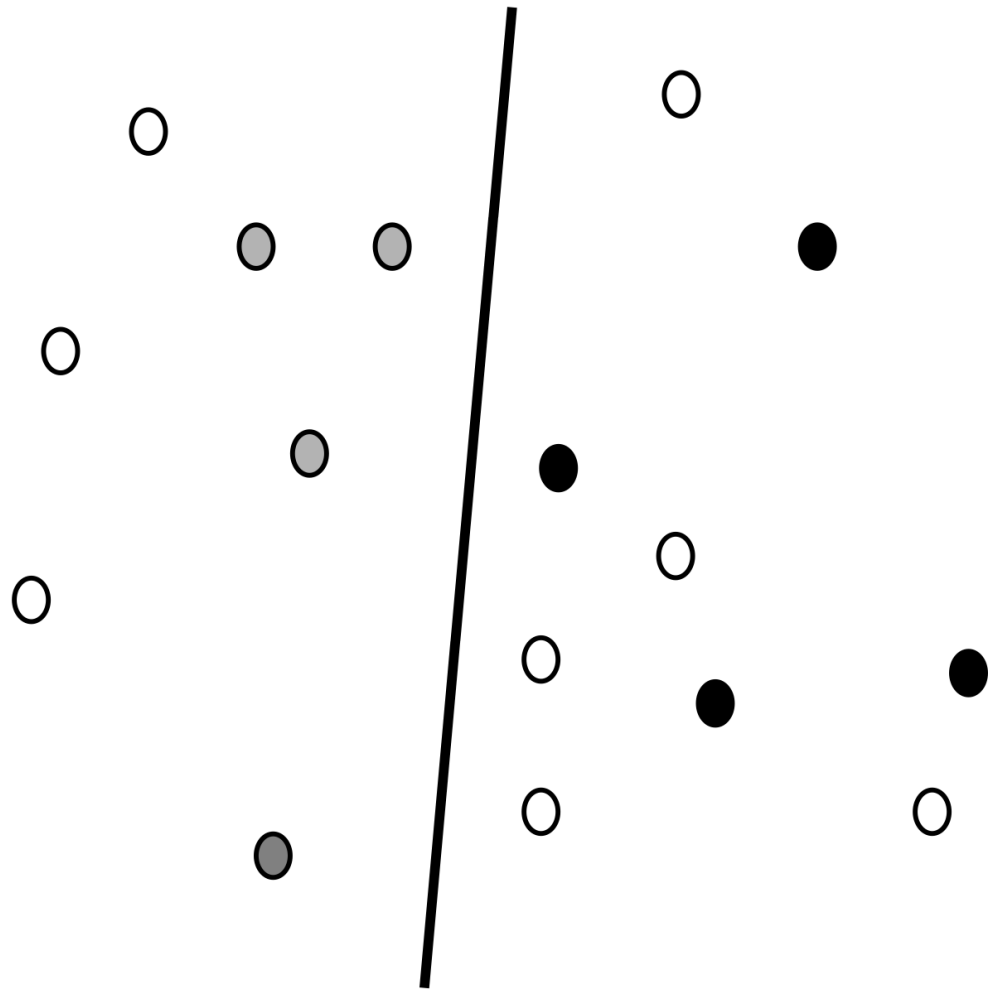
Fit a Classifier to Labeled Data



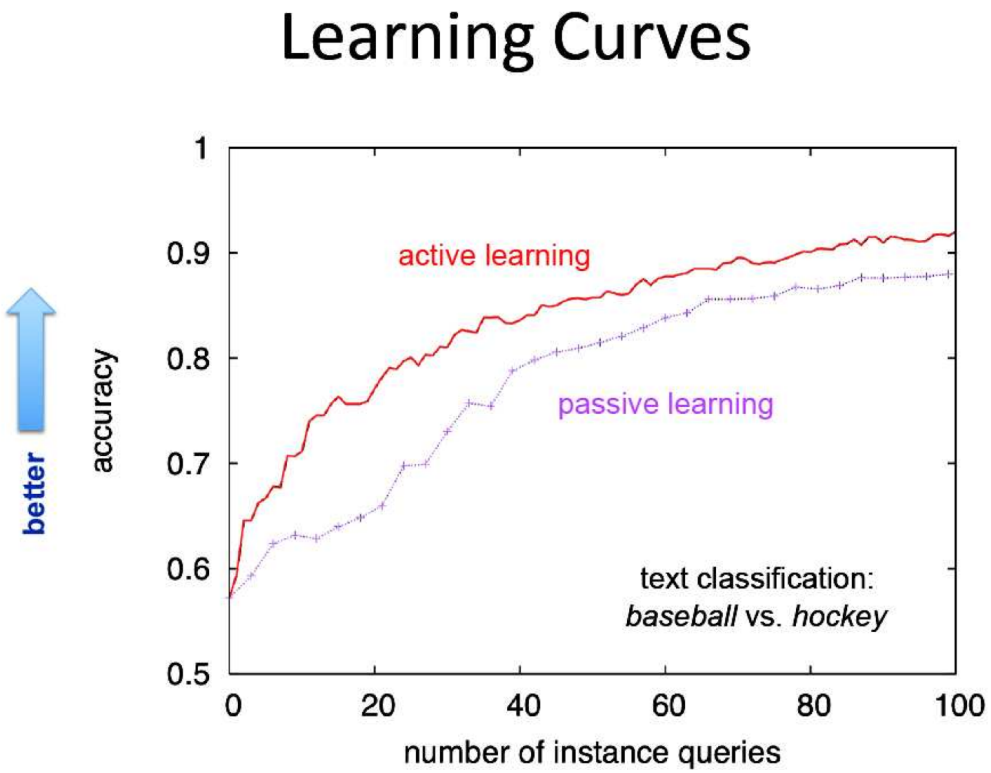
Pick the Best Next Point To Label



Fit a Classifier to Labeled Data



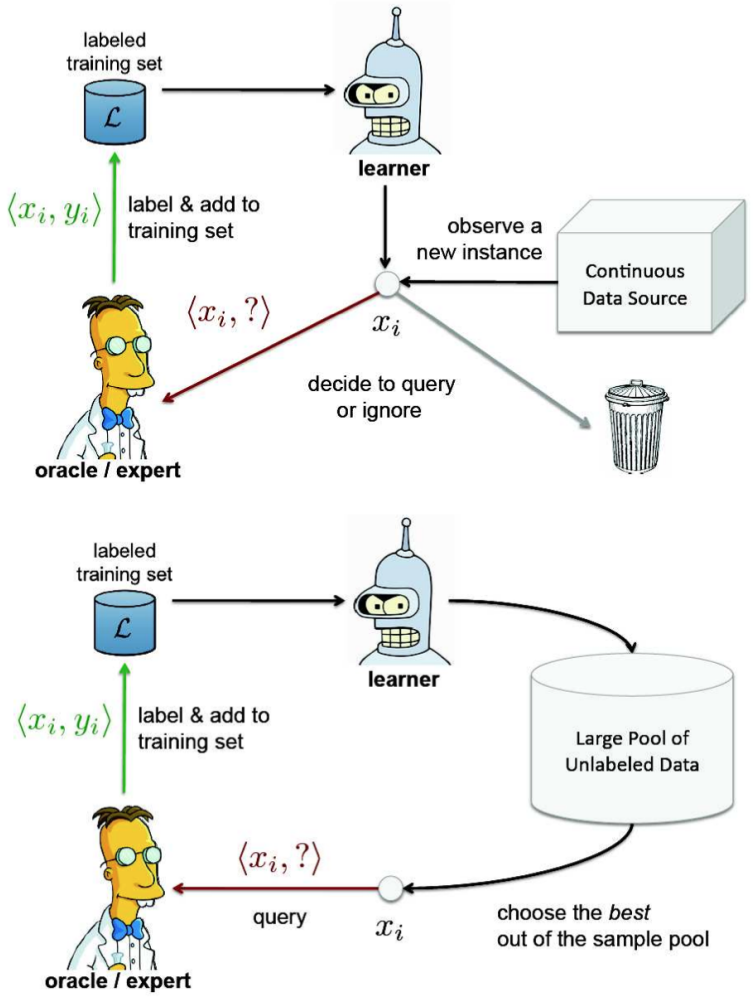
- Passive Learning curve: Randomly selects examples to get labels for
- Acctive learning curve: Active learning examples to get labels for



Types of Active Learning

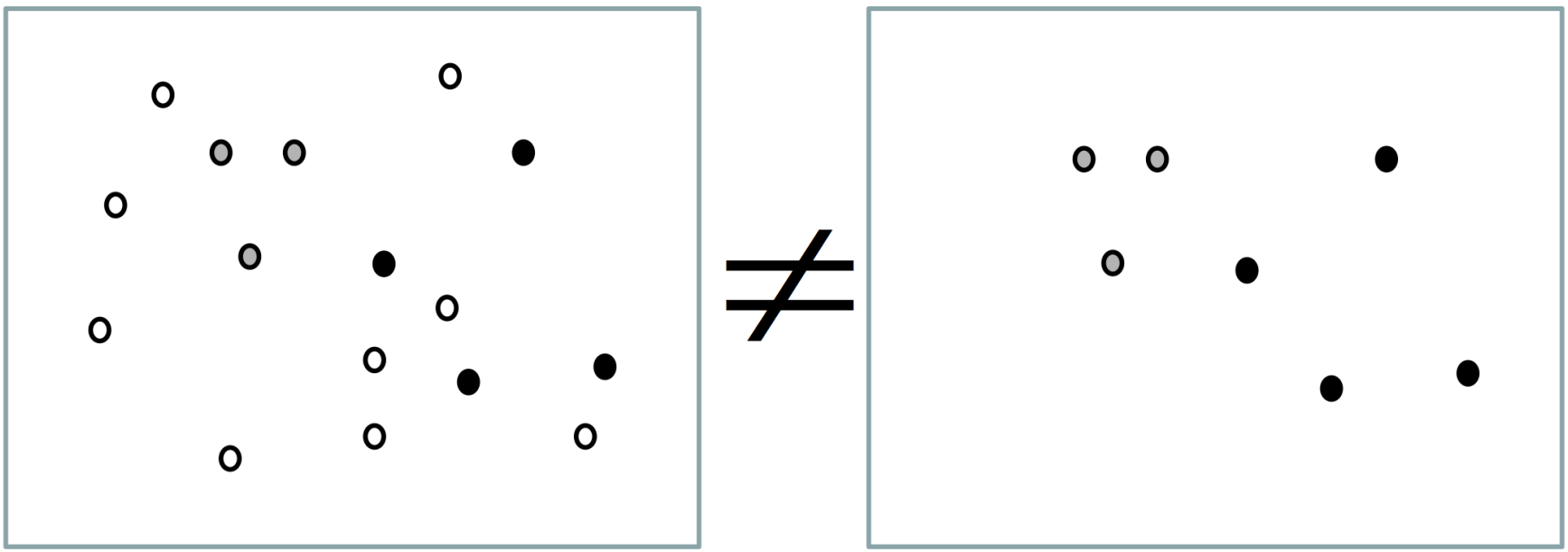
Largely falls into one of these two types:

- Stream-Based Active Learning
  - Consider one unlabeled example at a time
  - Decide whether to query its label or ignore it
- Pool-Based Active Learning
  - Given: a large unlabeled pool of examples
  - Rank examples in order of informativeness
  - Query the labels for the most informative example(s)



Biased Sampling

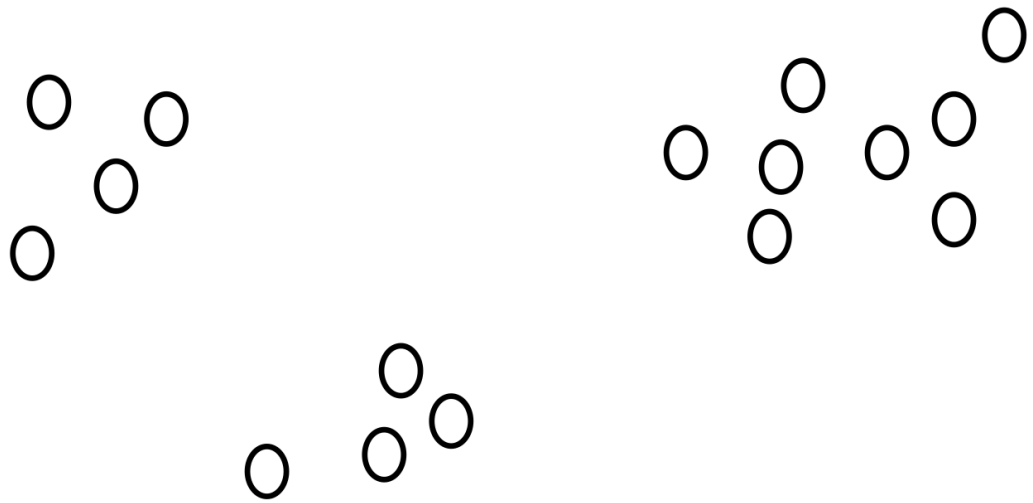
- The labeled points may not be representative of the underlying distribution
- This can increase error, even with infinitely many labeled samples



Two Rationales for Active Learning

- Rationale 1: We can exploit cluster structure in data
- Rationale 2: We can efficiently search through the hypothesis space

If the data looked like this ...



... then we might just need 3 labeled points

- Issues
  - Structure may not be so clearly defined
  - Structure exists at many levels of granularity
  - Clusters may not be all one label

Notations

- Problem: train hypothesis $h : \mathbb{X} \to Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\ldots}$ when getting labels y_i is expensive
- Input: initial labeled sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
- Output: hypothesis h and labeled sample $S_{m+k} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m+k}$
- General workflow:
  1. Train h using initial sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
  2. For all $i = m+1, \ldots, m+k$
     - select an object $\mathbf{x}_i$
     - get a label y_i
     - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

Uncertainty sampling

- Select examples which the current model is the most uncertain about
- Various ways to measure uncertainty. For example:
  - Based on the distance from the hyperplane
  - Using the label probability $P(y|\mathbf{x})$ (for probabilistic models)
- Let us consider multi-class classification based on some probabilistic model $P(y|\mathbf{x})$
- Decision according to the model is equal to

$$h(\mathbf{x}) = \arg\max_{y \in Y} P(y|\mathbf{x})$$

- Let us denote by $p_r(\mathbf{x})$, $r = 1, 2, \ldots, |Y|$ values of $P(y|\mathbf{x})$, $y \in Y$, ranked in decreasing order

Uncertainty sampling

- Least confidence principle

$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in\mathbb{X}} p_1(\mathbf{x})$$

- Margin sampling principle

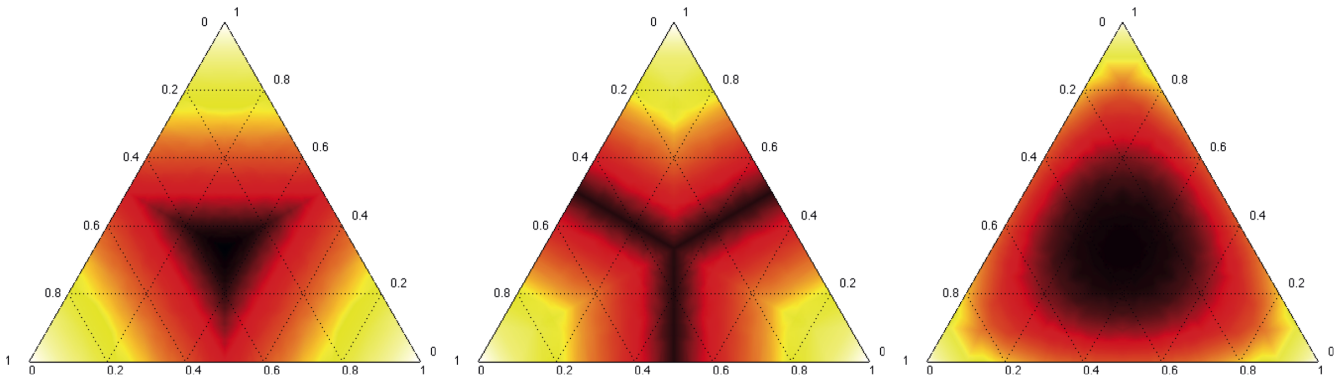
$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in\mathbb{X}} (p_1(\mathbf{x}) - p_2(\mathbf{x}))$$

- Maximum entropy principle

$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in\mathbb{X}} \sum_r p_r(\mathbf{x}) \log p_r(\mathbf{x})$$

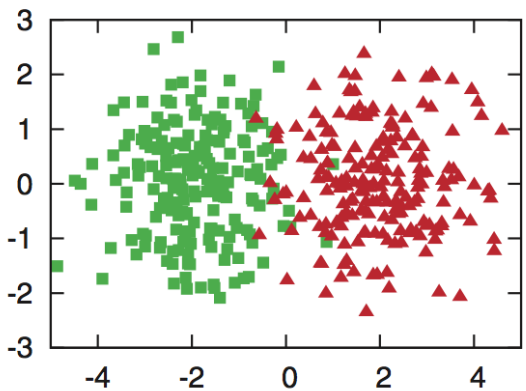
Uncertainty sampling

- In case of two classes these three principles are equivalent
- In a multi-class setting there are differences
- Below we show contour lines the corresponding criteria

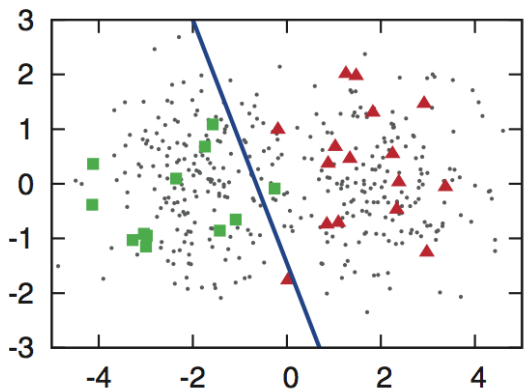


least confident margin entropy

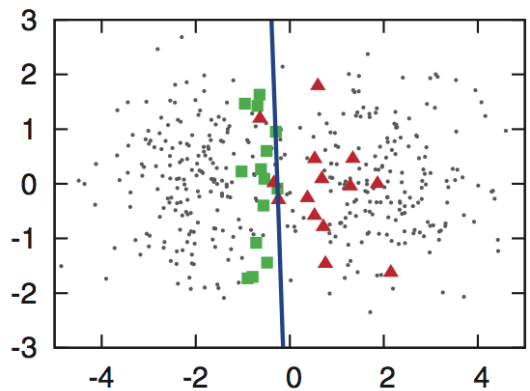
- Example. Synthetic dataset: $m = 30$, $m + k = 400$
  - two Gaussian density
  - logistic regression is constructed using 30 randomly selected objects
  - logistic regression is constructed using 30, adaptively selected using active learning



(a) a 2D toy data set



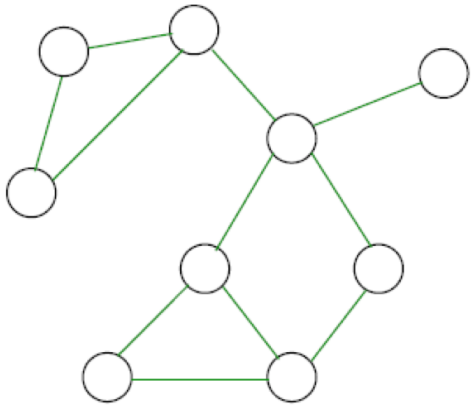
(b) random sampling



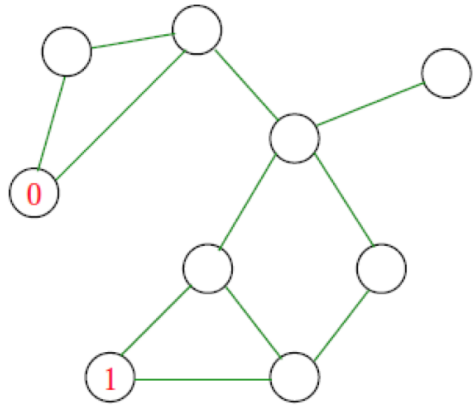
(c) uncertainty sampling

Uncertainty Sampling based on Label-Propagation

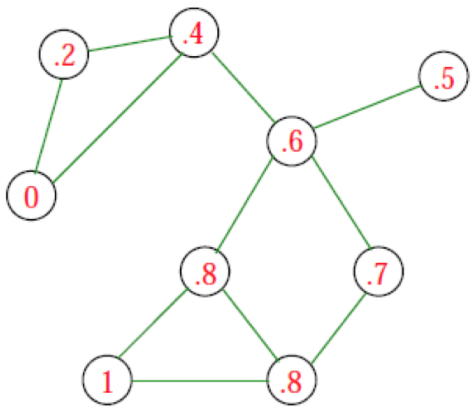
(1) Build neighborhood graph



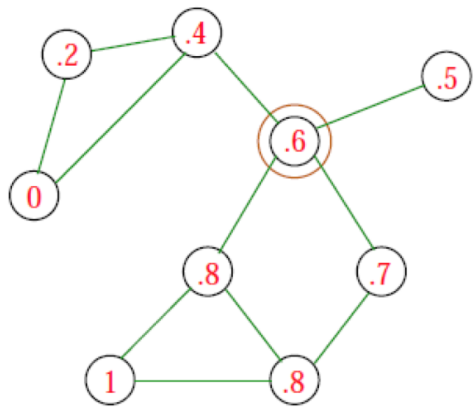
(2) Query some random points



(3) Propagate labels



(4) Make query and go to (3)



- Select $\mathbf{x}_i$ having the highest disagreement between decisions of a committee of models $h_t(\mathbf{x}_i) = \arg\max_{y \in Y} P_t(y|\mathbf{x})$, $t = 1, \ldots, T$
- Maximum Entropy Principle: select those $\mathbf{x}_i$ for which $h_t(\mathbf{x}_i)$ are the most different

$$\mathbf{x}_i = \arg\min_{\mathbf{z} \in X} \sum_{y \in Y} \widehat{p}(y|\mathbf{z}) \log \widehat{p}(y|\mathbf{z}),$$

where $\widehat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^{T} 1_{h_t(\mathbf{z})=y}$

- Maximum of an average KL-divergence principle: select those $\mathbf{x}_i$ on which $P_t(y|\mathbf{x}_i)$ are the most different

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{t=1}^{T} \mathrm{KL}\left(P_t(y|\mathbf{z}) \overline{P}(y|\mathbf{z})\right),$$

where $\overline{P}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^{T} P_t(y|\mathbf{z})$

Expected Model Change

- Select those $\mathbf{x}_i$, which would provide the highest change of a model
- We use a parametric classification model

$$h_\theta(\mathbf{x}) = \arg\max_{y \in Y} P(y|\mathbf{x}, \theta)$$

- For $\mathbf{z} \in \mathbb{X}$ and $y \in Y$ estimate a step of stochastic gradient descent when performing re-learning of the model with additional data point $(\mathbf{z}, y)$
- Let us denote by $\nabla_\theta \widehat{R}(\theta; \mathbf{z}, y)$ is a gradient vector the loss function
- We calculate

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{y \in Y} P(y|\mathbf{z}, \theta) \left\| \nabla_\theta \widehat{R}(\theta; \mathbf{z}, y) \right\|$$

Expected Error Reduction

- Select those $\mathbf{x}_i$, which after re-learning provides the most reliable classification of the non-labeled feature vectors from $\mathbb{X}$
- For $\mathbf{z} \in \mathbb{X}$ and $y \in Y$ we construct a classifier, adding to S_m additional example $(\mathbf{z}, y)$

$$h_{\mathbf{z}y}(\mathbf{x}) = \arg\max_{u \in Y} P_{\mathbf{z}y}(u|\mathbf{x})$$

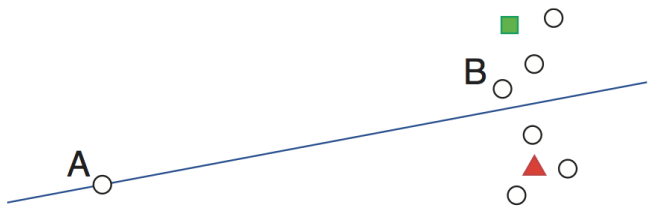
- Maximum Reliability of non-labeled data principle

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in \mathbb{X}} \sum_{y \in Y} P(y|\mathbf{z}) \sum_{j=m+1}^{m+k} P_{\mathbf{z}y}(h_{\mathbf{z}y}(\mathbf{x}_j)|\mathbf{x}_j)$$

- Minimum Entropy of non-labeled data principle

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in \mathbb{X}} \sum_{y \in Y} P(y|\mathbf{z}) \sum_{j=m+1}^{m+k} \sum_{u \in Y} P_{\mathbf{z}y}(u|\mathbf{x}_j) \log P_{\mathbf{z}y}(u|\mathbf{x}_j)$$

- Reduce weight of nonrepresentational objects
- Example: object A is more boundary, but it is less representative than B



- Any criterion for sampling has the form

$$\mathbf{x}_i = \arg\max_{\mathbf{x}} \phi(\mathbf{x})$$

- It can be adjusted by a local density estimate

$$\mathbf{x}_i = \arg\max_{\mathbf{x}} \phi(\mathbf{x}) \left(\sum_{j=m+1}^{m+k} \mathrm{sim}(\mathbf{x}, \mathbf{x}_j) \right)^{\beta},$$

where $\mathrm{sim}(\cdot, \cdot)$ is a some similarity function (the closer the bigger)

Example of Density Weighting