CSC 411 Lecture 19: Bayesian Linear Regression

Roger Grosse, Amir-massoud Farahmand, and Juan Carrasquilla

University of Toronto

Overview

- We've covered both parametric and nonparametric models for regression and classification.
  - Parametric models summarize the data into a finite-sized model. E.g., linear regression, logistic regression, neural nets, (linear) SVM, Naïve Bayes, GDA
  - Nonparametric models refer back to the data to make predictions. E.g., KNN
- The next two lectures are about Bayesian approaches to regression.
  - This lecture: Bayesian linear regression, a parametric model
  - Next lecture: Gaussian processes, a nonparametric model

Overview

- We're going to be Bayesian about the parameters of the model.
  - This is in contrast with naïve Bayes and GDA: in those cases, we used Bayes' rule to infer the class, but used point estimates of the parameters.
  - By inferring a posterior distribution over the *parameters*, the model can know what it doesn't know.
- How can uncertainty in the predictions help us?
  - Smooth out the predictions by averaging over lots of plausible explanations (just like ensembles!)
  - Assign confidences to predictions
  - Make more robust decisions
  - Guide exploration (focus on areas you're uncertain about)
    - E.g., Bayesian optimization (see next tutorial)

Recap: Linear Regression

- Given a training set of inputs and targets $\{(\mathbf{x}^{(i)}, t^{(i)})\}_{i=1}^{N}$
- Linear model:

$$y = \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x})$$

- Squared error loss:

$$\mathcal{L}(y, t) = \frac{1}{2}(t - y)^2$$

- L_2 regularization:

$$\mathcal{R}(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- Solution 1: solve analytically by setting the gradient to 0

$$\mathbf{w} = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Psi}^\top \mathbf{t}$$

- Solution 2: solve approximately using gradient descent

$$\mathbf{w} \leftarrow (1 - \alpha\lambda)\mathbf{w} - \alpha \boldsymbol{\Psi}^\top (\mathbf{y} - \mathbf{t})$$

Recap: Linear Regression

- We can give linear regression a probabilistic interpretation by assuming a Gaussian noise model:

$$t \,|\, \mathbf{x} \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}),\ \sigma^2)$$

- Linear regression is just maximum likelihood under this model:

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} \log p(t^{(i)} \,|\, \mathbf{x}^{(i)}; \mathbf{w}, b) &= \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(t^{(i)}; \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}), \sigma^2) \\
&= \frac{1}{N}\sum_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t^{(i)} - \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}))^2}{2\sigma^2}\right)\right] \\
&= \text{const} - \frac{1}{2N\sigma^2}\sum_{i=1}^{N} (t^{(i)} - \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}))^2
\end{aligned}$$

Recap: Linear Regression

- We can view an L_2 regularizer as MAP inference with a Gaussian prior.
- Recall MAP inference:

$$\arg\max_{\mathbf{w}} \log p(\mathbf{w} \mid \mathcal{D}) = \arg\max_{\mathbf{w}} \left[\log p(\mathbf{w}) + \log p(\mathcal{D} \mid \mathbf{w})\right]$$

- We just derived the likelihood term $\log p(\mathcal{D} \mid \mathbf{w})$:

$$\log p(\mathcal{D} \mid \mathbf{w}) = -\frac{1}{2N\sigma^2} \sum_{i=1}^{N} (t^{(i)} - \mathbf{w}^\top \mathbf{x} - b)^2 + \text{const}$$

- Assume a Gaussian prior, $\mathbf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$:

$$\begin{aligned}
\log p(\mathbf{w}) &= \log \mathcal{N}(\mathbf{w}; \mathbf{m}, \mathbf{S}) \\
&= \log \left[\frac{1}{(2\pi)^{D/2} |\mathbf{S}|^{1/2}} \exp\left(-\tfrac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) \right) \right] \\
&= -\tfrac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) + \text{const}
\end{aligned}$$

- Commonly, $\mathbf{m} = \mathbf{0}$ and $\mathbf{S} = \eta \mathbf{I}$, so

$$\log p(\mathbf{w}) = -\frac{1}{2\eta} \|\mathbf{w}\|^2 + \text{const.}$$

This is just L_2 regularization!

Recap: Full Bayesian Inference

- Recall: full Bayesian inference makes predictions by averaging over all likely explanations under the posterior distribution.
- Compute posterior using Bayes' Rule:

$$p(\mathbf{w} \mid \mathcal{D}) \propto p(\mathbf{w}) p(\mathcal{D} \mid \mathbf{w})$$

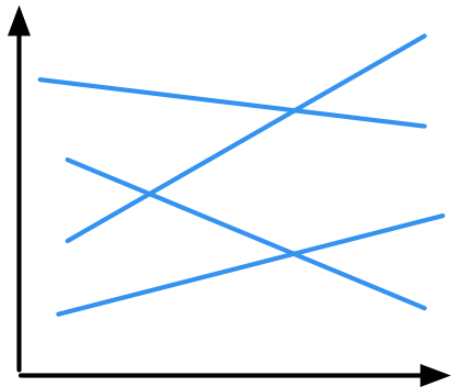
- Make predictions using the posterior predictive distribution:

$$p(t \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{w} \mid \mathcal{D})\, p(t \mid \mathbf{x}, \mathbf{w})\, \mathrm{d}\mathbf{w}$$

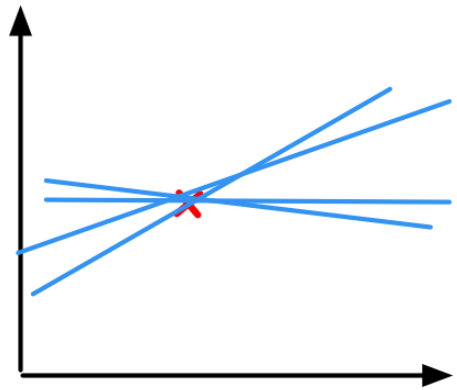
- Doing this lets us quantify our uncertainty.

Bayesian Linear Regression

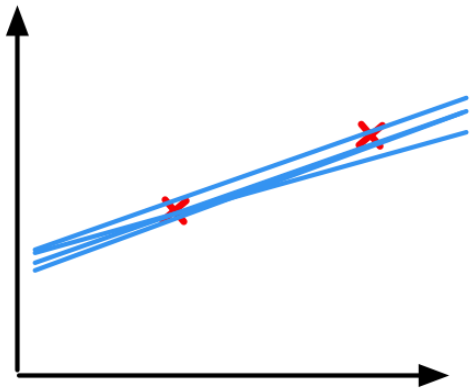
- Bayesian linear regression considers various plausible explanations for how the data were generated.
- It makes predictions using all possible regression weights, weighted by their posterior probability.



no observations



one observation



two observations

- **Prior distribution:** $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{S})$
- **Likelihood:** $t \,|\, \mathbf{x}, \mathbf{w} \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}),\ \sigma^2)$
- Assuming fixed/known $\mathbf{S}$ and σ^2 is a big assumption. More on this later.

Bayesian Linear Regression: Posterior

- **Deriving the posterior distribution:**

$$\begin{aligned}
\log p(\mathbf{w} \mid \mathcal{D}) &= \log p(\mathbf{w}) + \log p(\mathcal{D} \mid \mathbf{w}) + \text{const} \\
&= -\tfrac{1}{2}\mathbf{w}^\top \mathbf{S}^{-1}\mathbf{w} - \frac{1}{2\sigma^2}\|\boldsymbol{\Psi}\mathbf{w} - \mathbf{t}\|^2 + \text{const} \\
&= -\tfrac{1}{2}\mathbf{w}^\top \mathbf{S}^{-1}\mathbf{w} - \frac{1}{2\sigma^2}\left(\mathbf{w}^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi}\mathbf{w} - 2\mathbf{t}^\top \boldsymbol{\Psi}\mathbf{w} + \mathbf{t}^\top \mathbf{t}\right) + \text{const} \\
&= -\tfrac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{w} - \boldsymbol{\mu}) + \text{const} \qquad \text{(complete the square!)}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top \mathbf{t} \\
\boldsymbol{\Sigma}^{-1} &= \sigma^{-2}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \mathbf{S}^{-1}
\end{aligned}$$

- This is a multivariate Gaussian distribution, i.e.

$$\mathbf{w} \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Bayesian Linear Regression: Posterior

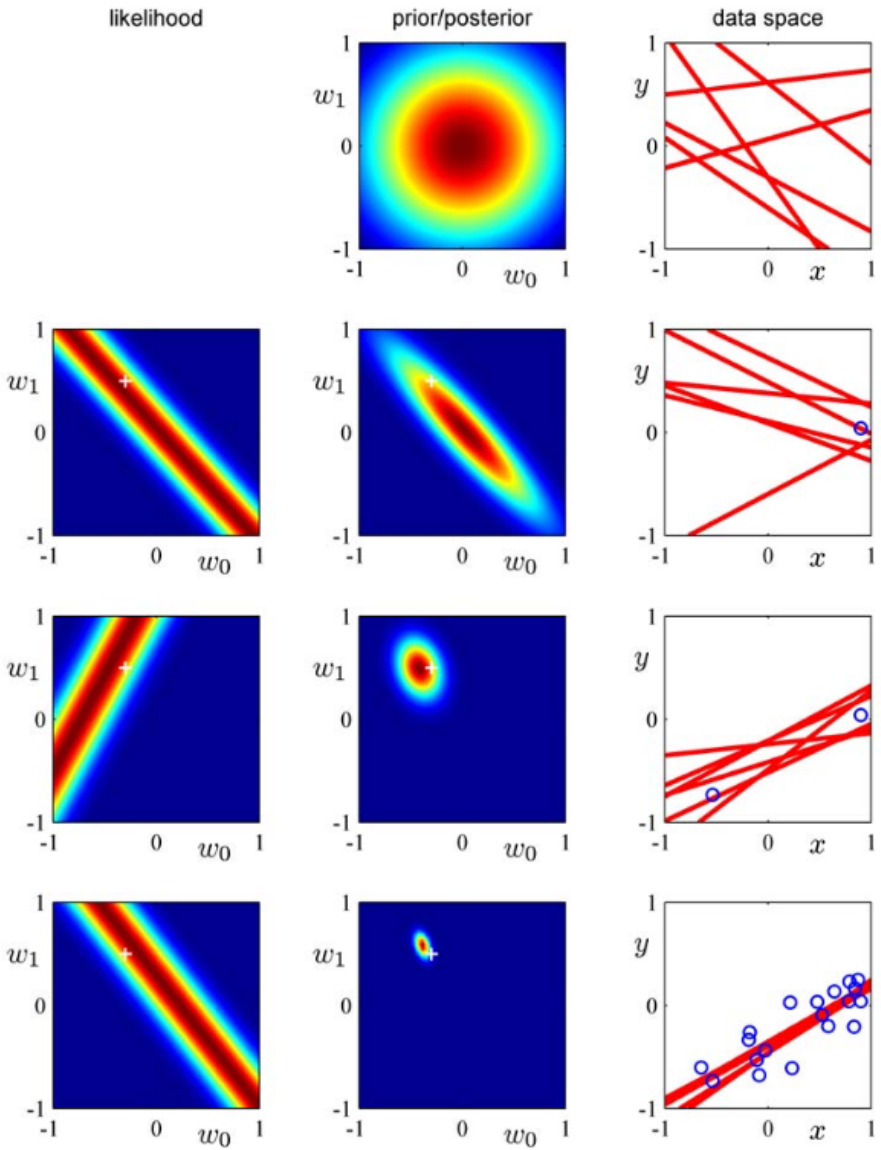
- Just showed:

$$\mathbf{w} \,|\, \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Psi}^\top \mathbf{t}$$
$$\boldsymbol{\Sigma}^{-1} = \sigma^{-2} \boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \mathbf{S}^{-1}$$

- Since a Gaussian prior leads to a Gaussian posterior, this means the Gaussian distribution is the conjugate prior for linear regression!
- Compare $\boldsymbol{\mu}$ the closed-form solution for linear regression:

$$\mathbf{w} = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Psi}^\top \mathbf{t}$$

Bayesian Linear Regression

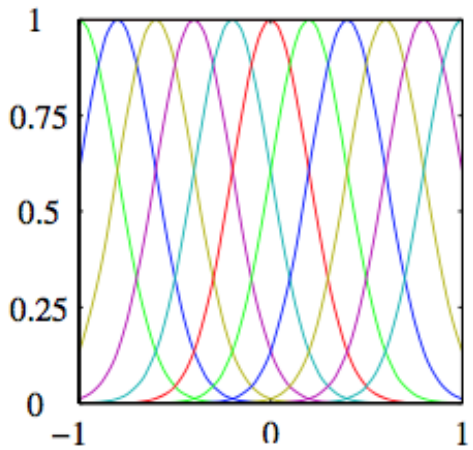


— Bishop, Pattern Recognition and Machine Learning

Bayesian Linear Regression

- Example with radial basis function (RBF) features

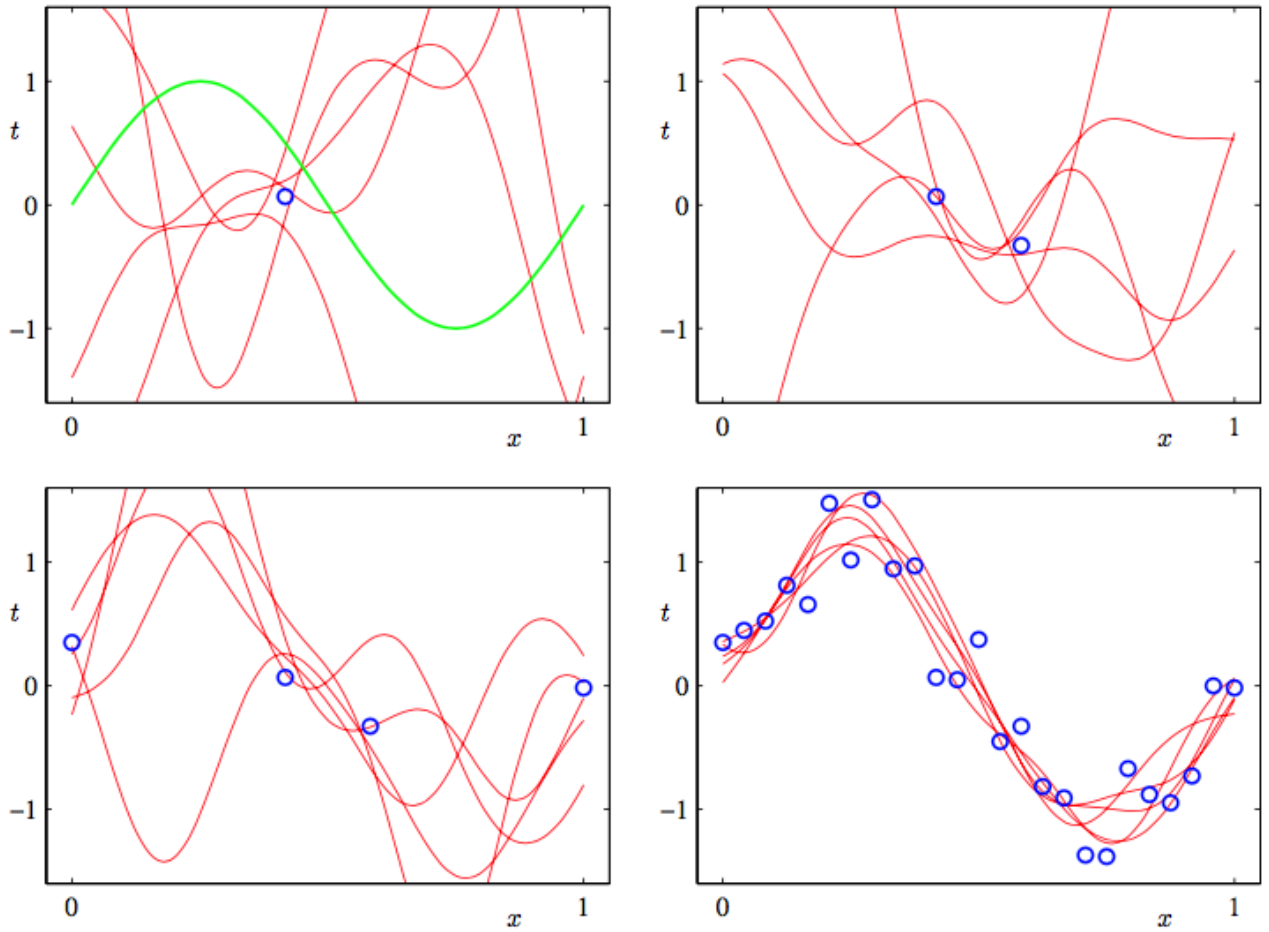
$$\psi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$



— Bishop, Pattern Recognition and Machine Learning

Bayesian Linear Regression

Functions sampled from the posterior:



— Bishop, Pattern Recognition and Machine Learning

Bayesian Linear Regression

- Posterior predictive distribution:

$$p(t \mid \mathbf{x}, \mathcal{D}) = \int \underbrace{p(t \mid \mathbf{x}, \mathbf{w})}_{\mathcal{N}(t\,;\,\mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}),\sigma)} \underbrace{p(\mathbf{w} \mid \mathcal{D})}_{\mathcal{N}(\mathbf{w}\,;\,\boldsymbol{\mu},\boldsymbol{\Sigma})} \, \mathrm{d}\mathbf{w}$$

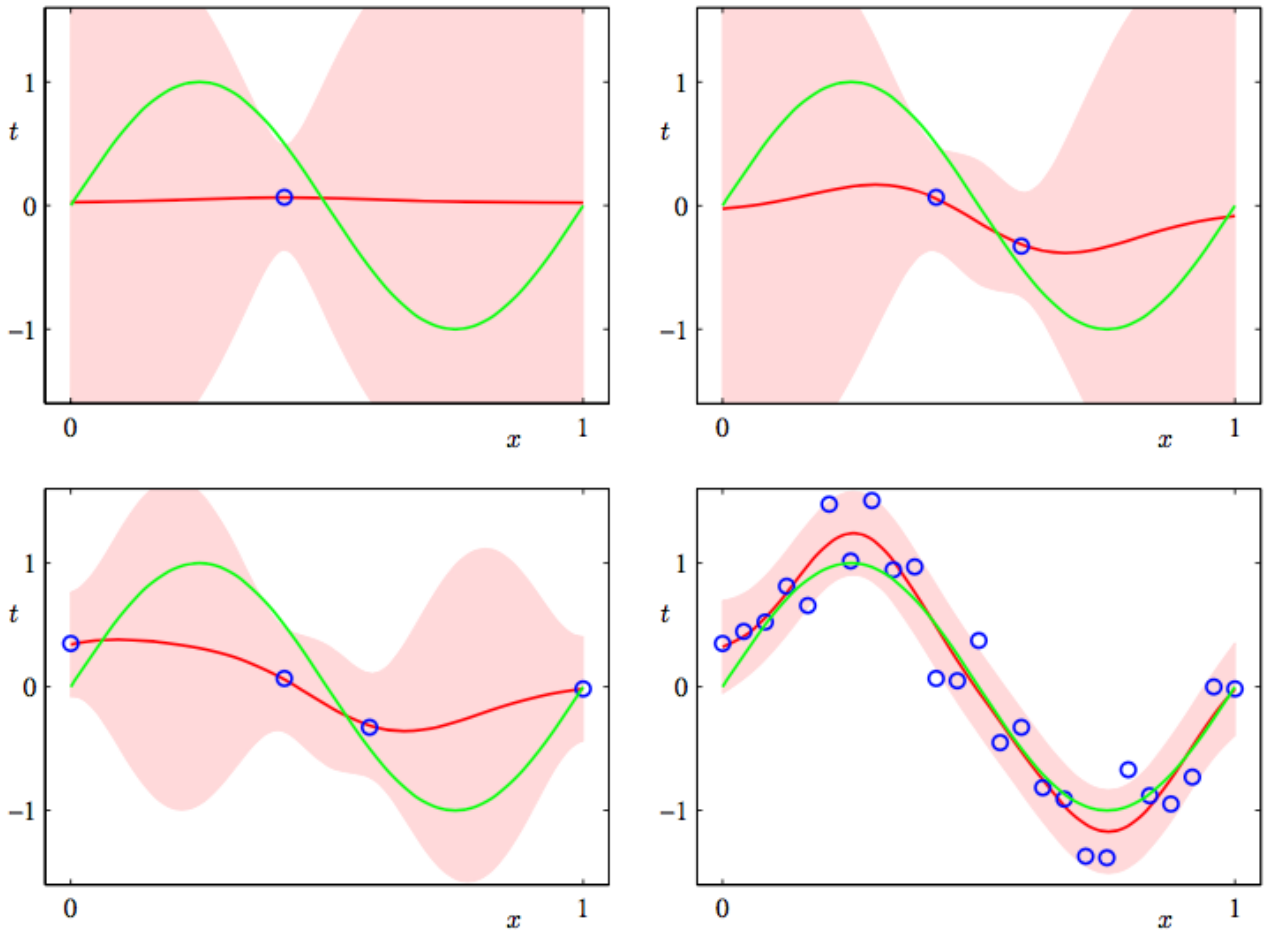
- Another interpretation: $t = \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma)$ is independent of $\mathbf{w}$.
- By the linear combination rules for Gaussian random variables, t is a Gaussian distribution with parameters

$$\mu_{\text{pred}} = \boldsymbol{\mu}^\top \boldsymbol{\psi}(\mathbf{x})$$
$$\sigma^2_{\text{pred}} = \boldsymbol{\psi}(\mathbf{x})^\top \boldsymbol{\Sigma} \boldsymbol{\psi}(\mathbf{x}) + \sigma^2$$

- Hence, the posterior predictive distribution is $\mathcal{N}(t\,;\,\mu_{\text{pred}}, \sigma^2_{\text{pred}})$.

Bayesian Linear Regression

Here we visualize confidence intervals based on the posterior predictive mean and variance at each point:



— Bishop, Pattern Recognition and Machine Learning

Bayesian Decision Theory

- What do we actually do with the posterior predictive distribution $p(t \mid \mathbf{x}, \mathcal{D})$?
- Often, we want to make a decision. We can formulate this as minimizing the expected loss under the posterior distribution. This is known as decision theory.
- Simple example: want to choose a single prediction y to minimize the expected squared error loss.

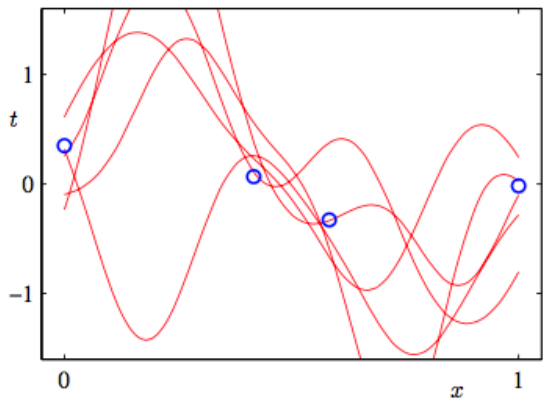
$$\arg\min_y \mathbb{E}_{p(t \mid \mathbf{x}, \mathcal{D})}[(y - t)^2] = \mathbb{E}_{p(t \mid \mathbf{x}, \mathcal{D})}[t]$$

  - Same derivation as bias/variance from Lecture 4
  - Similarly, you can show that under absolute value loss, you should pick the median.

Now for a more interesting use of Bayesian decision theory...

Bayesian Optimization

- Black-box optimization: we want to minimize a function, but we only get to query function values (i.e. no gradients!)
  - Each query is expensive, so we want to do as few as possible
  - Canonical example: minimize the validation error of an ML algorithm with respect to its hyperparameters
- Bayesian Optimization: approximate the function with a simpler function (e.g. linear in a feature representation), called the surrogate function.
- After we've queried a certian number of points, we can condition on these to infer the posterior over the surrogate function using Bayesian linear regression.



Bayesian Optimization

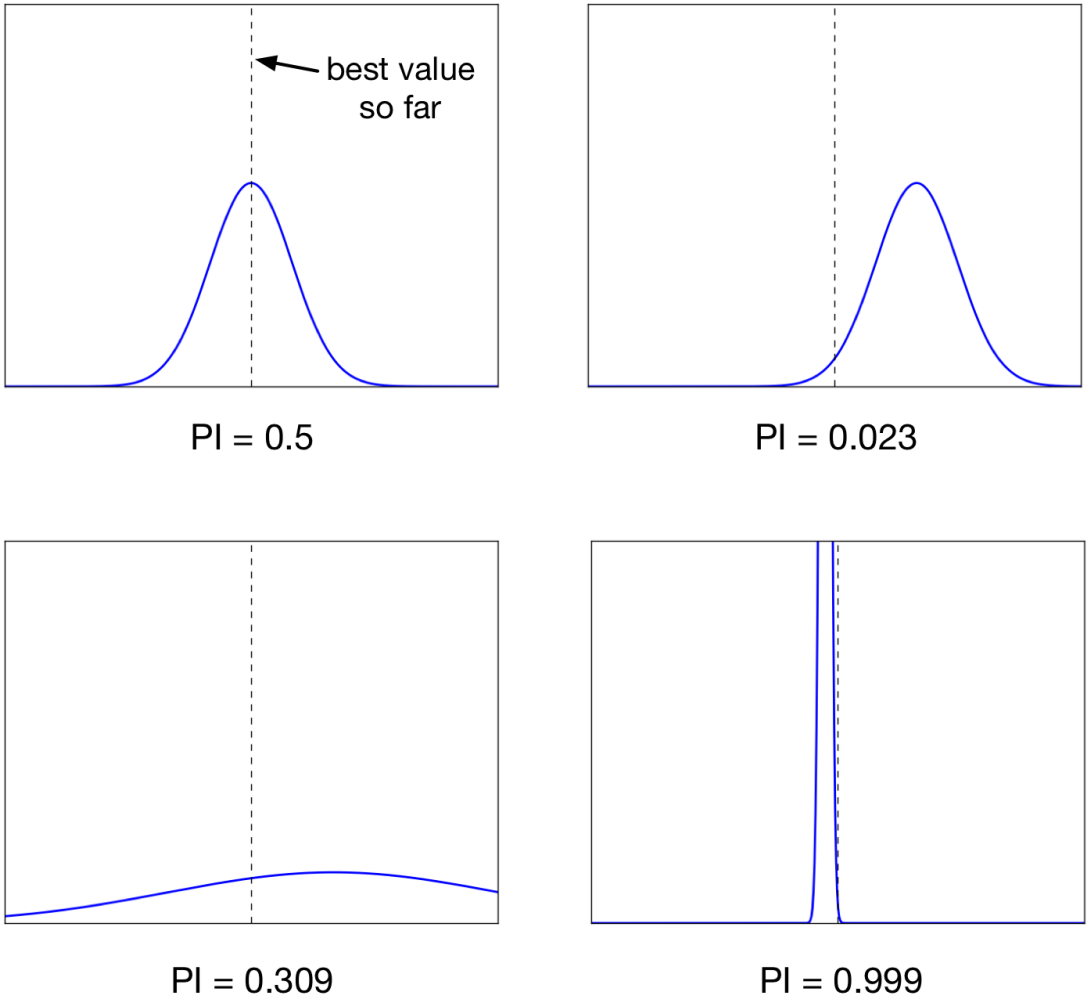
- To choose the next point to query, we must define an acquisition function, which tells us how promising a candidate it is.
- What's wrong with the following acquisition functions:
  - posterior mean: $-\mathbb{E}[f(\theta)]$
  - posterior variance: $\mathrm{Var}(f(\theta))$
- Desiderata:
  - high for points we expect to be good
  - high for points we're uncertain about
  - low for points we've already tried
- Candidate 1: probability of improvement (PI)

$$\mathrm{PI} = \Pr(f(\theta) < \gamma - \epsilon),$$

where γ is the best value so far, and ϵ is small.

Bayesian Optimization

Examples:



- Plots show the posterior predictive distribution for $f(\theta)$.

Bayesian Optimization

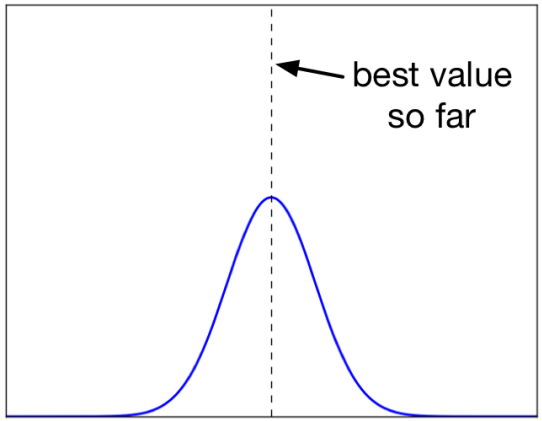
- The problem with Probability of Improvement (PI): it queries points it is highly confident will have a small imporvement
  - Usually these are right next to ones we've already evaluated
- A better choice: Expected Improvement (EI)

$$\mathrm{EI} = \mathbb{E}[\max(\gamma - f(\theta), 0)]$$

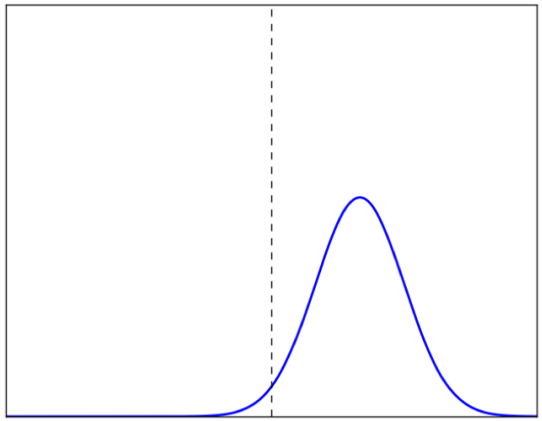
  - The idea: if the new value is much better, we win by a lot; if it's much worse, we haven't lost anything.
  - There is an explicit formula for this if the posterior predictive distribution is Gaussian.

Bayesian Optimization

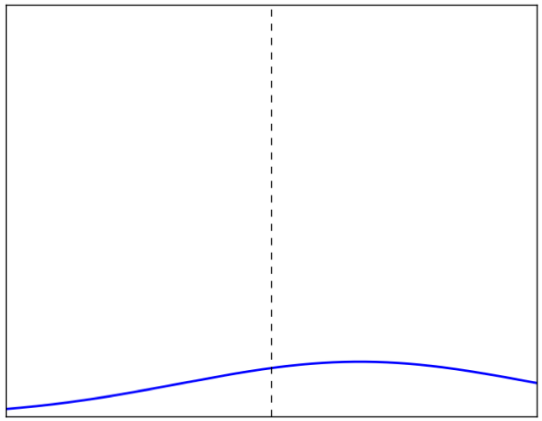
Examples:



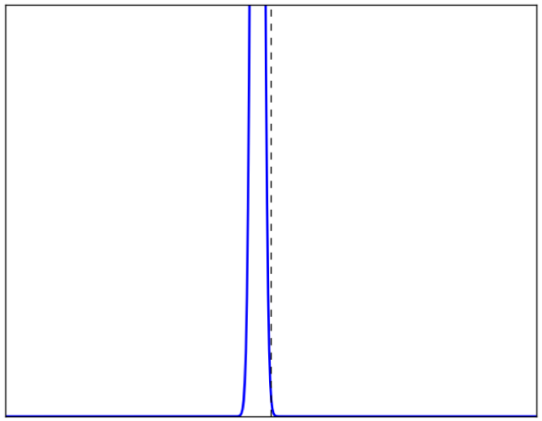
EI = 0.199



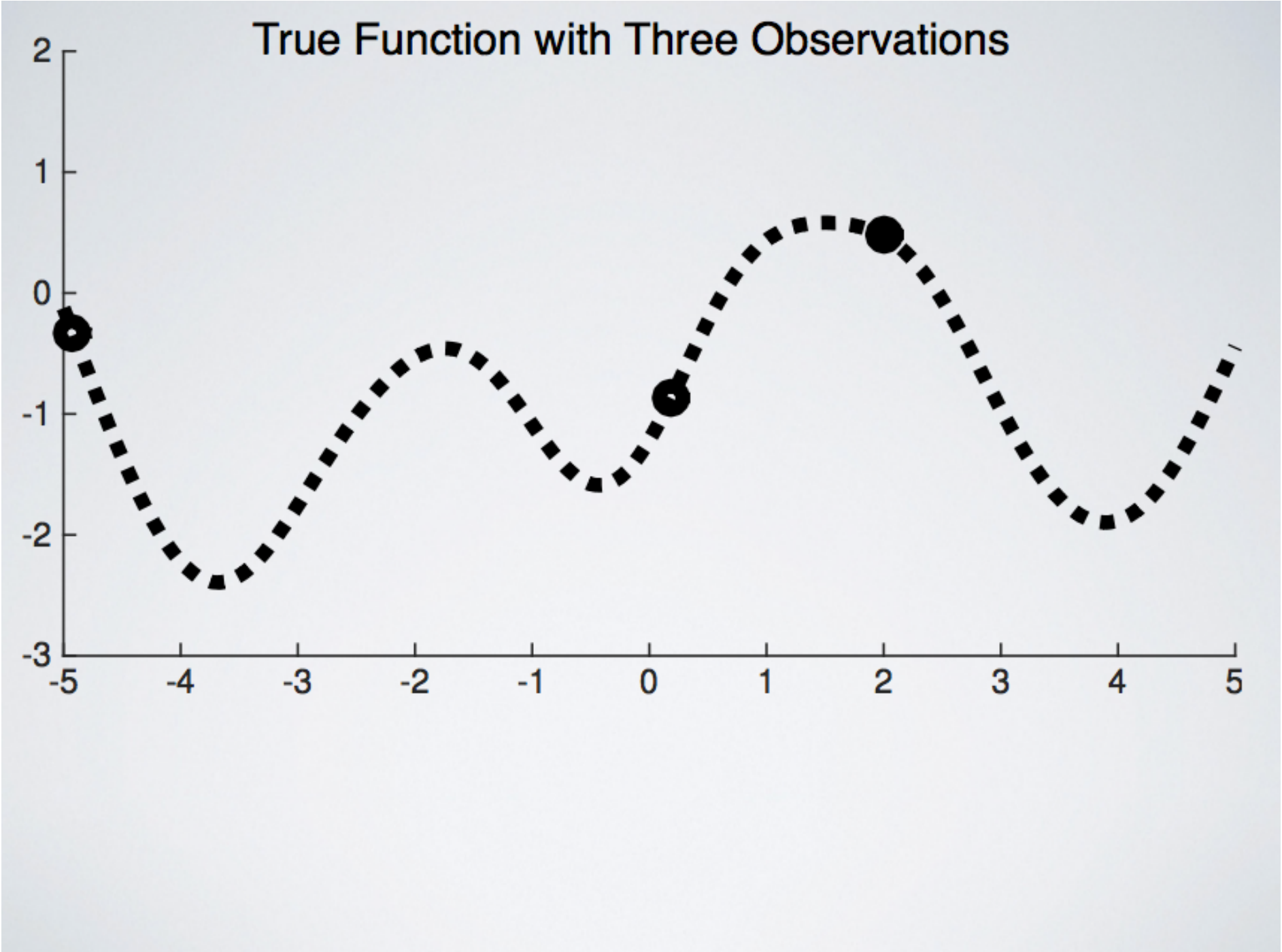
EI = 0.004

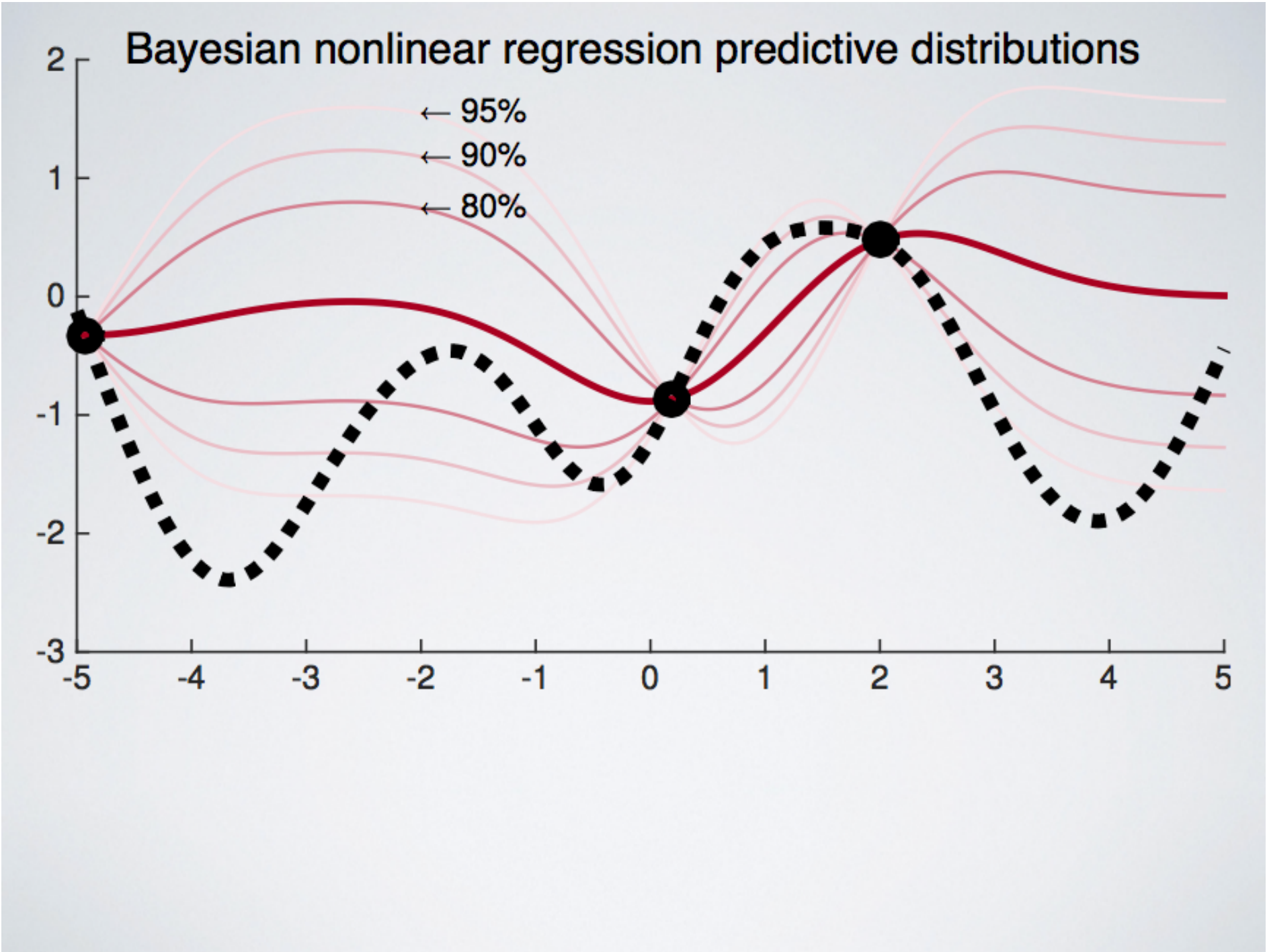


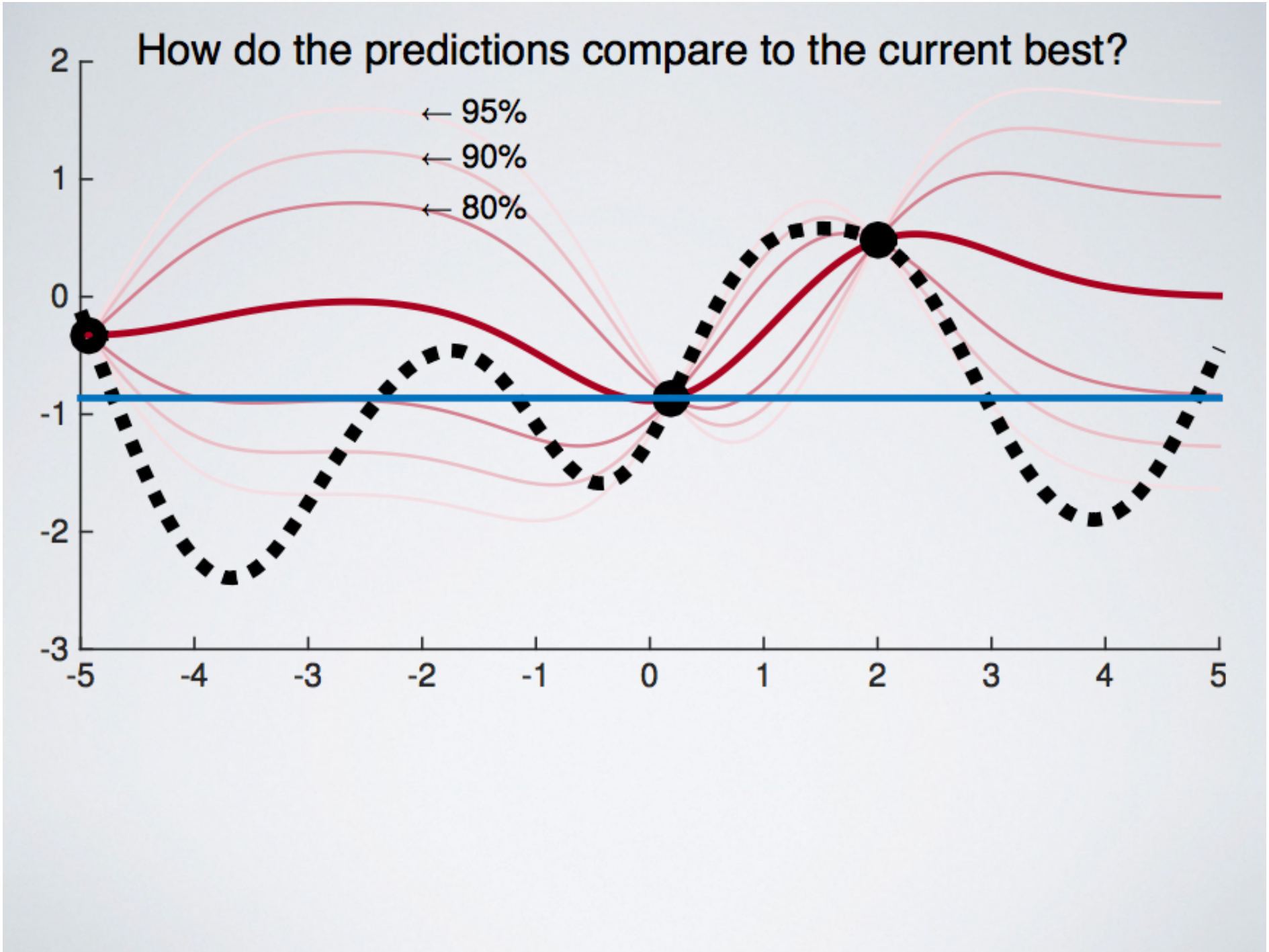
EI = 0.396

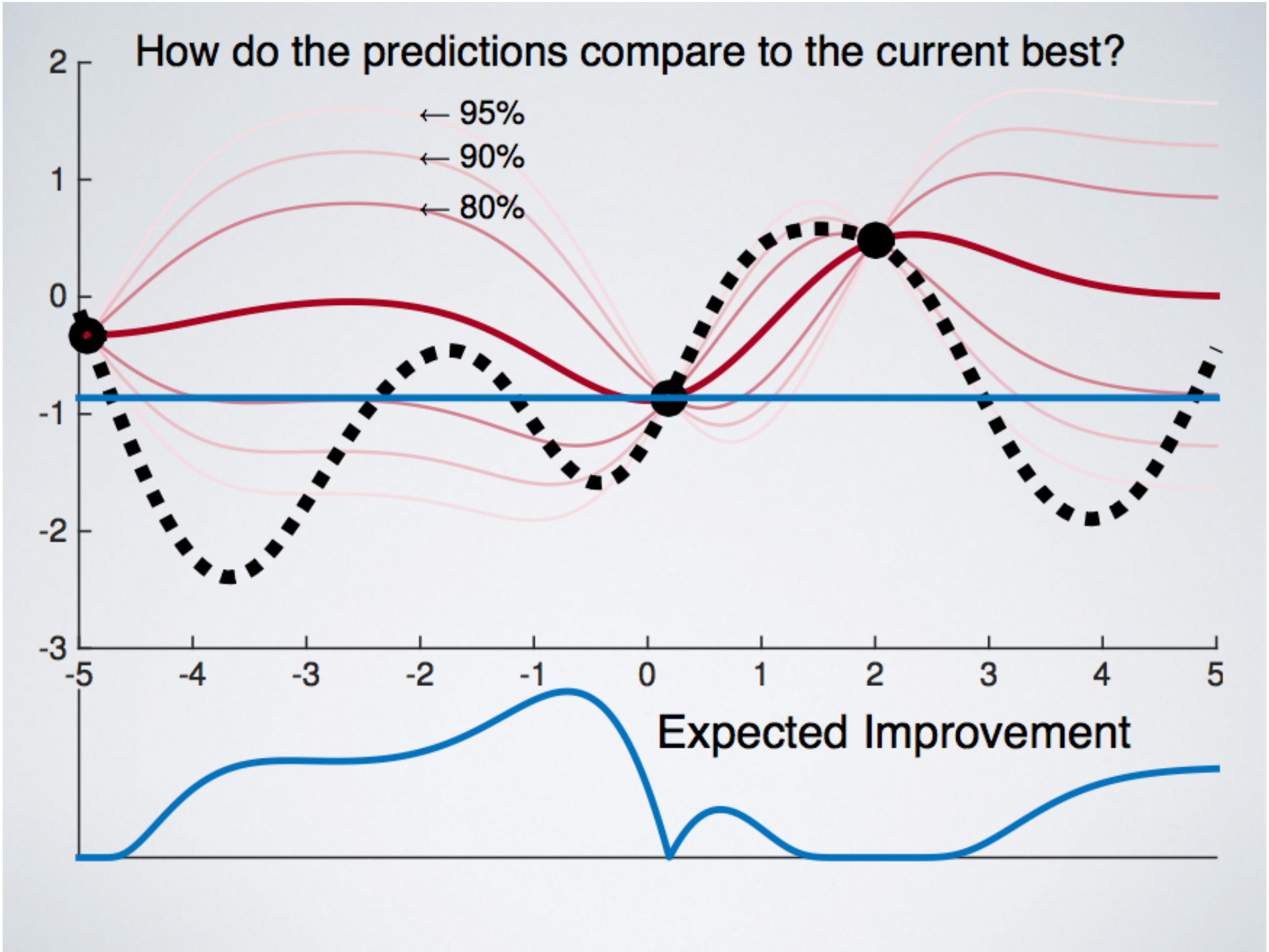


EI = 0.15


True Function with Three Observations
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5


Bayesian nonlinear regression predictive distributions
← 95%
← 90%
← 80%
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5


How do the predictions compare to the current best?
← 95%
← 90%
← 80%
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5

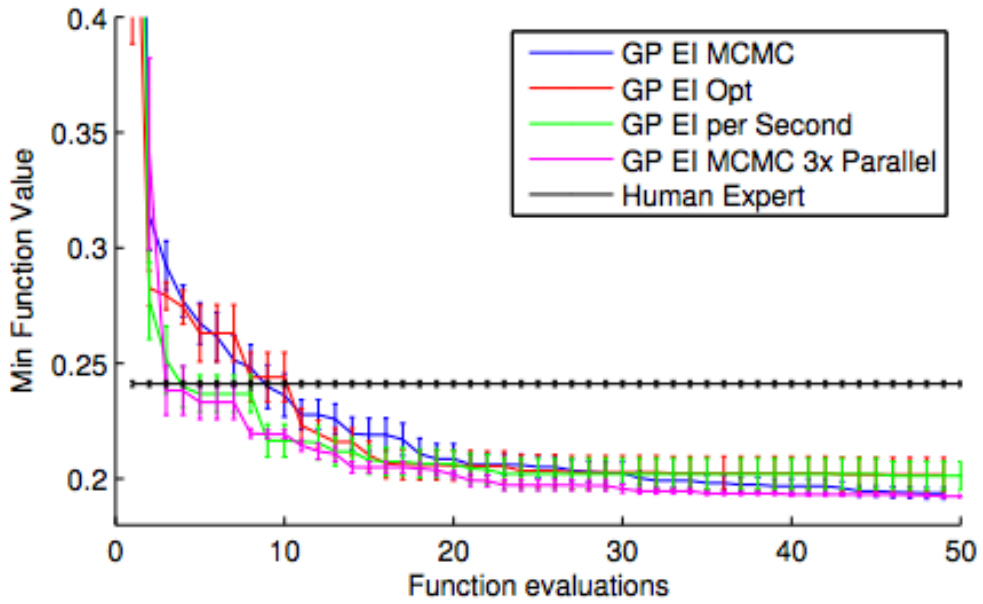

How do the predictions compare to the current best?
← 95%
← 90%
← 80%
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
Expected Improvement

Bayesian Optimization

- I showed one-dimensional visualizations, but the higher-dimensional case is conceptually no different.
  - Maximize the acquisition function using gradient descent
  - Use lots of random restarts, since it is riddled with local maxima
  - BayesOpt can be used to optimize tens of hyperparameters.
- I've described BayesOpt in terms of Bayesian linear regression with basis functions learned by a neural net.
  - In practice, it's typically done with Gaussian processes, which are the topic of next lecture.
  - But Bayesian linear regression is actually useful, since it scales better to large numbers of queries.
- One variation: some configurations can be much more expensive than others
  - Use another Bayesian regression model to estimate the computational cost, and query the point that maximizes expected improvement per second

Bayesian Optimization

- BayesOpt can often beat hand-tuned configurations in a relatively small number of steps.
- Results on optimizing hyperparameters (layer-specific learning rates, weight decay, and a few other parameters) for a CIFAR-10 conv net:



- Each function evaluation takes about an hour
- Human expert = Alex Krizhevsky, the creator of AlexNet

Optional material

Occam's Razor (optional)

- Occam's Razor: "Entities should not be multiplied beyond necessity."
  - Named after the 14th century British theologian William of Occam
- Huge number of attempts to formalize mathematically
  - See Domingos, 1999, "The role of Occam's Razor in knowledge discovery" for a skeptical overview.

    `https://homes.cs.washington.edu/~pedrod/papers/dmkd99.pdf`
- Common misinterpretation: your prior should favor simple explanations

Occam's Razor (optional)

- Suppose you have a finite set of models, or hypotheses $\{\mathcal{H}_i\}_{i=1}^M$ (e.g. polynomials of different degrees)
- Posterior inference over models (Bayes' Rule):

$$p(\mathcal{H}_i \mid \mathcal{D}) \propto \underbrace{p(\mathcal{H}_i)}_{\text{prior}} \underbrace{p(\mathcal{D} \mid \mathcal{H}_i)}_{\text{evidence}}$$

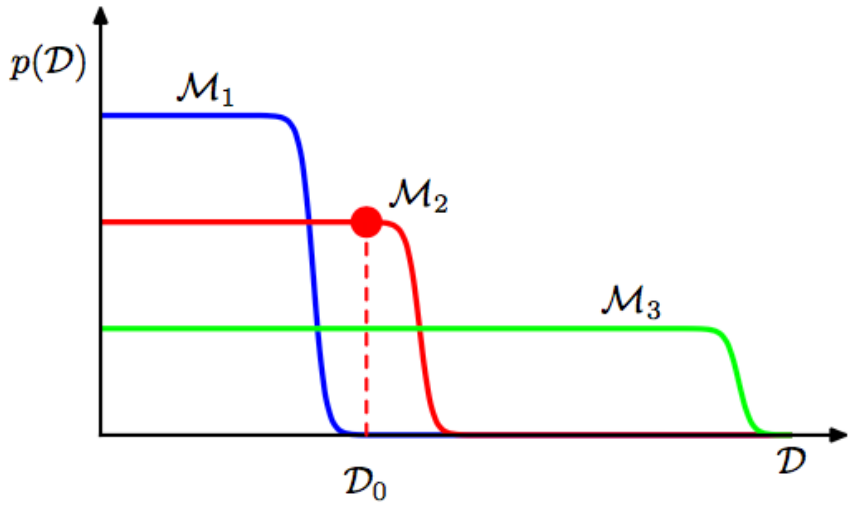
- Which of these terms do you think is more important?
- The evidence is also called marginal likelihood since it requires marginalizing out the parameters:

$$p(\mathcal{D} \mid \mathcal{H}_i) = \int p(\mathbf{w} \mid \mathcal{H}_i)\, p(\mathcal{D} \mid \mathbf{w}, \mathcal{H}_i)\, \mathrm{d}\mathbf{w}$$

- If we're comparing a handful of hypotheses, $p(\mathcal{H}_i)$ isn't very important, so we can compare them based on marginal likelihood.

Occam's Razor (optional)

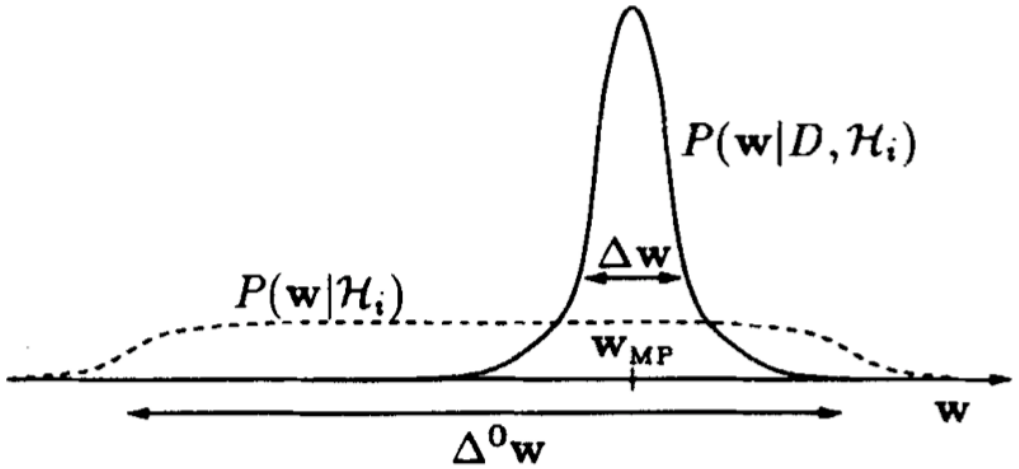
- Suppose M_1, M_2, and M_3 denote a linear, quadratic, and cubic model.
- M_3 is capable of explaning more datasets than M_1.
- But its distribution over $\mathcal{D}$ must integrate to 1, so it must assign lower probability to ones it can explain.



— Bishop, Pattern Recognition and Machine Learning

Occam's Razor (optional)

- How does the evidence (or marginal likelihood) penalize complex models?



- Approximating the integral:

$$p(\mathcal{D} \mid \mathcal{H}_i) = \int p(\mathcal{D} \mid \mathbf{w}, \mathcal{H}_i)\, p(\mathbf{w} \mid \mathcal{H}_i)$$

$$\simeq \underbrace{p(\mathcal{D} \mid \mathbf{w}_{\mathrm{MAP}}, \mathcal{H}_i)}_{\text{best-fit likelihood}} \underbrace{p(\mathbf{w}_{\mathrm{MAP}} \mid \mathcal{H}_i)\, \Delta\mathbf{w}}_{\text{Occam factor}}$$

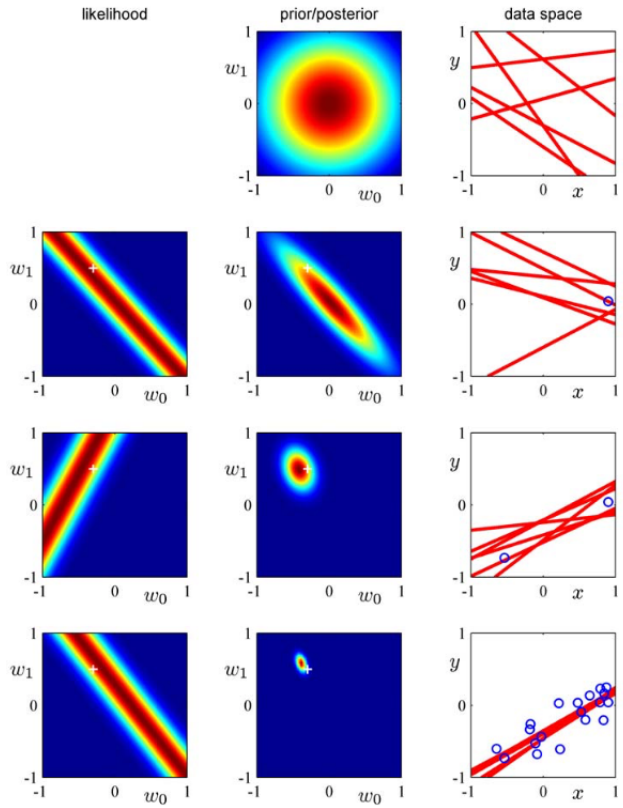
Occam's Razor (optional)

- Multivariate case:

$$p(\mathcal{D} \mid \mathcal{H}_i) \simeq \underbrace{p(\mathcal{D} \mid \mathbf{w}_{\mathrm{MAP}}, \mathcal{H}_i)}_{\text{best-fit likelihood}} \underbrace{p(\mathbf{w}_{\mathrm{MAP}} \mid \mathcal{H}_i)\, |\mathbf{A}|^{-1/2}}_{\text{Occam factor}},$$

  where $\mathbf{A} = \nabla^2_{\mathbf{w}} \log p(\mathcal{D} \mid \mathbf{w}, \mathcal{H}_i)$

- The determinant appears because we're taking the volume.
- The more parameters in the model, the higher dimensional the parameter space, and the faster the volume decays.



— Bishop, Pattern Recognition and Machine Learning

Occam's Razor (optional)

- Analyzing the asymptotic behavior:

$$
\begin{aligned}
\mathbf{A} &= \nabla^2_{\mathbf{w}} \log p(\mathcal{D} \mid \mathbf{w}, \mathcal{H}_i) \\
&= \sum_{j=1}^{N} \underbrace{\nabla^2_{\mathbf{w}} \log p(y_i \mid \mathbf{x}_i, \mathbf{w}, \mathcal{H}_i)}_{\triangleq A_i} \\
&\approx N \, \mathbb{E}[A_i] \\
\log \text{ Occam factor} &= \log p(\mathbf{w}_{\text{MAP}} \mid \mathcal{H}_i) + \log |\mathbf{A}|^{-1/2} \\
&\approx \log p(\mathbf{w}_{\text{MAP}} \mid \mathcal{H}_i) + \log |N \, \mathbb{E}[A_i]|^{-1/2} \\
&= \log p(\mathbf{w}_{\text{MAP}} \mid \mathcal{H}_i) - \frac{1}{2} \log |\mathbb{E}[A_i]| - \frac{D \log N}{2} \\
&= \text{const} - \frac{D \log N}{2}
\end{aligned}
$$

- Bayesian Information Criterion (BIC): penalize the complexity of your model by $\frac{1}{2} D \log N$.

Occam's Razor (optional)

- Summary

$$p(\mathcal{H}_i \,|\, \mathcal{D}) \propto p(\mathcal{H}_i)\, p(\mathcal{D} \,|\, \mathcal{H}_i)$$
$$p(\mathcal{D} \,|\, \mathcal{H}_i) \simeq p(\mathcal{D} \,|\, \mathbf{w}_{\mathrm{MAP}}, \mathcal{H}_i)\, p(\mathbf{w}_{\mathrm{MAP}} \,|\, \mathcal{H}_i)\, |\mathbf{A}|^{-1/2}$$

  Asymptotically, with lots of data, this behaves like

$$\log p(\mathcal{D} \,|\, \mathcal{H}_i) = \log p(\mathcal{D} \,|\, \mathbf{w}_{\mathrm{MAP}}, \mathcal{H}_i) - \frac{1}{2} D \log N.$$

- Occam's Razor is about integration, not priors (over hypotheses).