A Bayesian View to Frequentist Problems

Bahman Moraffah
Electrical, Computer, and Energy Engineering
Arizona State University

https://bmoraffa.github.io/presentations

Ordinary Least Square

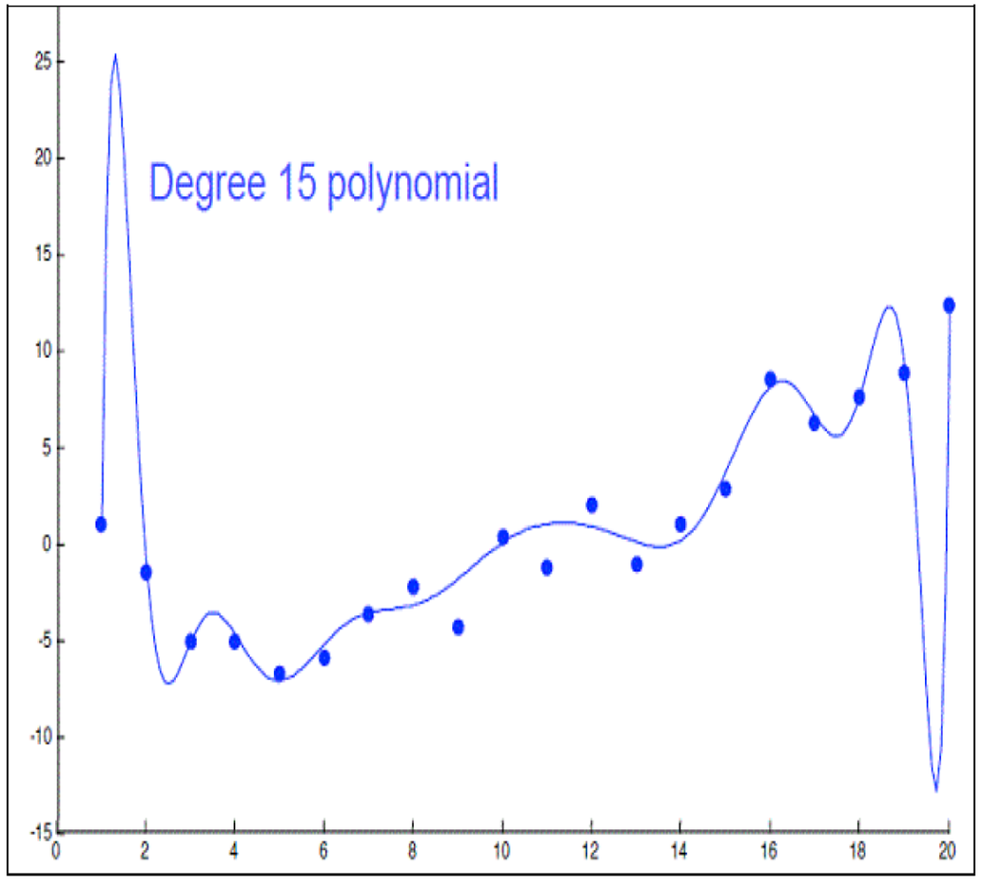
- The OLS estimates have relatively low bias and low variability especially when the relationship between the response and predictors is linear and n >> p.

- If n is not much larger than p, then the OLS fit can have high variance and may result in over fitting and poor estimates on unseen observations.

- If p > n, then the variability of the OLS fit increases dramatically, and the variance of these estimates in infinite.

Model Accuracy

- Carefully selected features can improve model accuracy but adding too many can lead to overfitting.
  - Overfitted models describe random error or noise instead of any underlying relationship.
  - They generally have poor predictive performance on test data.



- For instance, we can use a 15-degree polynomial function to fit the following data so that the fitted curve goes nicely through the data points.
- However, a brand new dataset collected from the same population may not fit this particular curve well at all.

Feature Selection

- Subset Selection
  - Identify a subset of the p predictors that we believe to be related to the response; then, fit a model using OLS on the reduced set.
  - Methods: best subset selection, stepwise selection

- Shrinkage (Regularization)
  - Involves shrinking the estimated coefficients toward zero relative to the OLS estimates; has the effect of reducing variance and performs variable selection.
  - Methods: ridge regression, lasso

- Dimension Reduction
  - Involves projecting the p predictors into a M-dimensional subspace, where M < p, and fit the linear regression model using the M projections as predictors.
  - Methods: principal components regression, partial least squares

Subset Selection: Shrinkage

- The subset selection methods use OLS to fit a linear model that contains a subset of the predictors.
- As an alternative, we can fit a model containing all p predictors using a technique that constrains or regularizes the coefficient estimates (i.e. shrinks the coefficient estimates towards zero).
- It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance.
- Regularization is our first weapon to combat overfitting.

Ridge Regression

- Recall that the OLS fitting procedure estimates the beta coefficients using the values that minimize:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- Ridge regression is similar to OLS, except that the coefficients are estimated by minimizing a slightly different quantity:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $\lambda \geq 0$ is a tuning parameter, to be determined separately.

Ridge Regression

- Note that $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage.
- The idea of penalizing by the sum-of-squares of the parameters is also used in neural networks, where it is known as weight decay.
- An equivalent way to write the ridge problem is:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t,$$

Tuning Parameter

where the tuning parameter λ is a positive value.

- This has the effect of shrinking the estimated beta coefficients towards zero. It turns out that such a constraint should improve the fit, because shrinking the coefficients can significantly reduce their variance.

- Note that when $\lambda = 0$, the penalty term as no effect, and ridge regression will procedure the OLS estimates. Thus, selecting a good value for λ is critical (can use cross-validation for this).

The Lasso

- One significant problem of ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.

- Thus, the final model will include all p predictors, which creates a challenge in model interpretation

- A more modern machine learning alternative is the lasso.

- The lasso works in a similar way to ridge regression, except it uses a different penalty term that shrinks some of the coefficients exactly to zero.

The Lasso

- The lasso coefficients minimize the quantity:

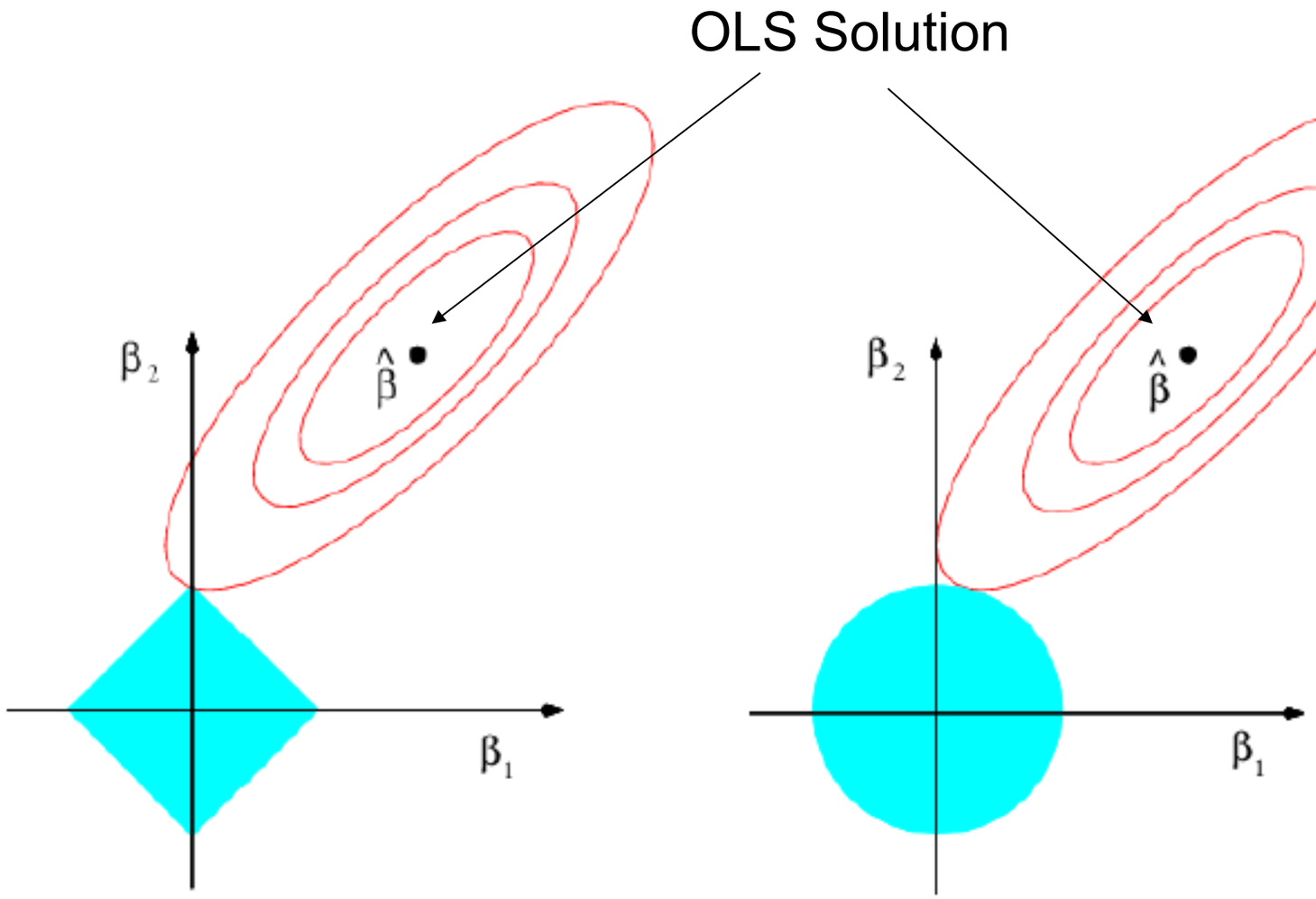
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| \;=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

- The key difference from ridge regression is that the lasso uses an ℓ_1 penalty instead of an ℓ_2, which has the effect of forcing some of the coefficients to be exactly equal to zero when the tuning parameter λ is sufficiently large.

- Thus, the lasso performs variable/feature selection.

The Lasso vs Ridge

- The lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region.



Lasso

Ridge Regression

How to select the Tuning Parameter

- Select a grid of potential values; use cross-validation to estimate the error rate on test data (for each value of λ) and select the value that gives the smallest error rate.
  - Finally, the model is re-fit using all of the variable observations and the selected value of the tuning parameter λ.
- Bayesian Optimization

Bayesian Statistics

- Bayesian Statistics:
  - Probabilistic modeling to express all forms of uncertainty and noise
  - … then *inverse probability* rule (i.e. Bayes' Theorem) allows us to infer unknown quantities, learn from data, and make predictions
    - Bayes' theorem:

$$Q(d\theta|X = x) = \frac{dP(X \in \cdot|\theta)}{dP(x \in \cdot)}Q(d\theta)$$

- Bayesian statistics that is not parametric (wait!)
- Bayesian nonparametrics (i.e. not finite parameter, unbounded/ growing/infinite number of parameters)
  - BNP models do not generally satisfy Bayes' theorem since the density cannot exist for all x (undominated models) (not the same as posterior tractability!)
  - Random discrete measures are often undominated.

Bayesian Nonparametric Model

- Why Bayesian nonparametrics?
  - Bayesian : Simplicity (of the framework)
  - Nonparametric : Complexity (of the real world phenomena)
- Definition: A BNP model is a Bayesian Model with an infinite-dimension parameter space and assumes that data distribution cannot be represented in terms of finite et of parameters.

Modeling Goal	Example process
Distribution on functions	Gaussian processes
Distribution on distributions	Dirichlet processes
	PY processes
Clustering	CRP / Polya Urn
Hierarchical clustering	Dirichlet diffusion tree
	Kingman's coalescent
Distribution on measures	Completely random measure
…	

Bayesian Lasso

- Reminder:

$$\min_{\boldsymbol{\beta}} (\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})^{\top} (\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

For the linear regression model

$$\mathbf{y} = \mu \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

Bayesian Statistics suggests to have priors:

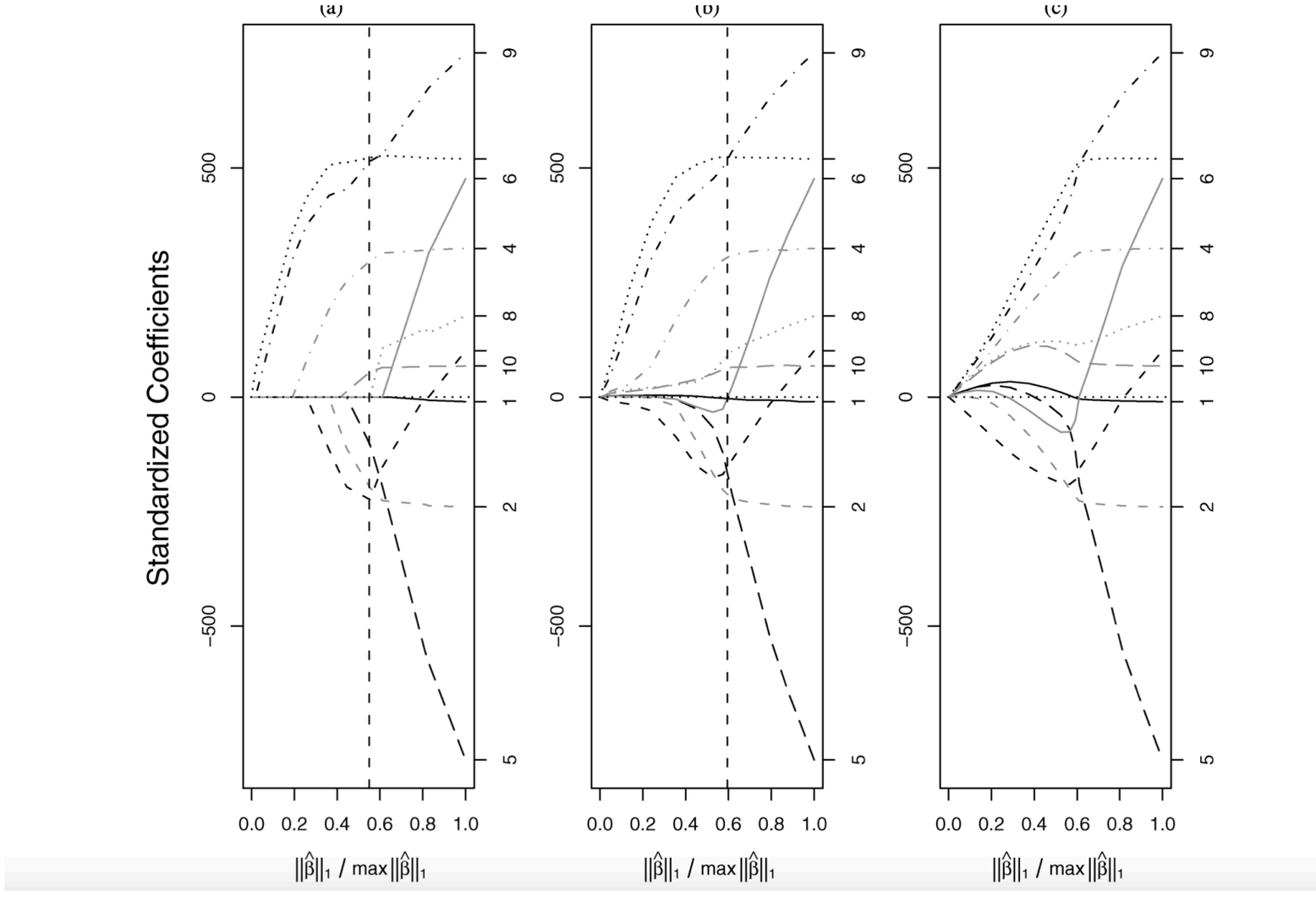
$$\pi(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}}$$

$$\pi(\sigma^2) = \frac{1}{\sigma^2}$$

Comparison

- The Bayesian Lasso estimates appear to be a compromise between the Lasso and ridge regression estimates; the paths are smooth, like ridge regression, but are more similar in shape to the Lasso paths, particularly when the L1 norm is relatively small.
- Bayesian Lasso appears to pull the more weakly related parameters to 0 faster than ridge regression does, indicating a potential advantage of the Laplace prior over a Gaussian (or a Student-t) prior.

Comparison



Lasso(a),Bayesian Lasso(b),and ridge regression(c)trace plots for estimates of the diabetes data regression parameters versus the relative L1 norm, with vertical lines for the Lasso and Bayesian Lasso indicating the estimates chosen by n-fold cross-validation and marginal maximum likelihood. The Bayesian Lasso estimates were posterior medians computed over a grid of λ values, using 10,000 consecutive iterations of the Gibbs sampler of Section 2 (after 1,000 burn-in iterations) for each λ.

Search for Good Parameters

- Define an objective function.
  - Most often, we care about generalization performance. Use cross validation to measure parameter quality!
- How do people currently search? Black magic.
  - Grid search
  - Random search

Painful!

- Requires many training cycles.
- Possibly Noisy.

Can We Do Better? Bayesian Optimization

- Build a probabilistic model for the objective. Include hierarchical structure about units, etc.
- Compute the posterior predictive distribution.
  Integrate out all the possible true functions.
  We use Gaussian process regression.
- Optimize a cheap proxy function instead.
  The model is much cheaper than that true objective.

The main insight!

Make the proxy function exploit uncertainty to balance exploration against exploitation.

Using Uncertainty in Optimization

- Find the minimum $x^* = argmax_{x \in \mathcal{X}} f(x)$
- We only can compute the function pointwise but do not have the functional form or the gradient (function behaves like a black box).
- After performing some evaluations, the GP gives us easy closed-form marginal mean and variances.
- Exploration: Seek places with high variance.
- Exploitation: Seek places with low mean.
- The acquisition function balances these for our proxy optimization to determine the next evaluation.

Bayesian Optimization uses all of the information from previous evaluations and performs some computation to determine the next point to try

Intuition: Bayesian Optimization

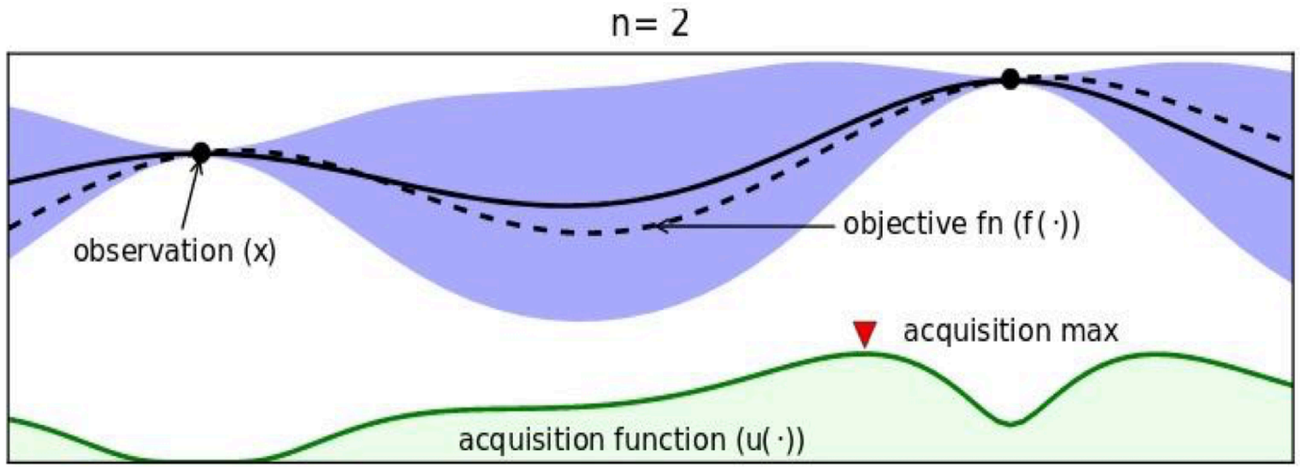
The main insight!

Make the proxy function exploit uncertainty to balance exploration against exploitation.

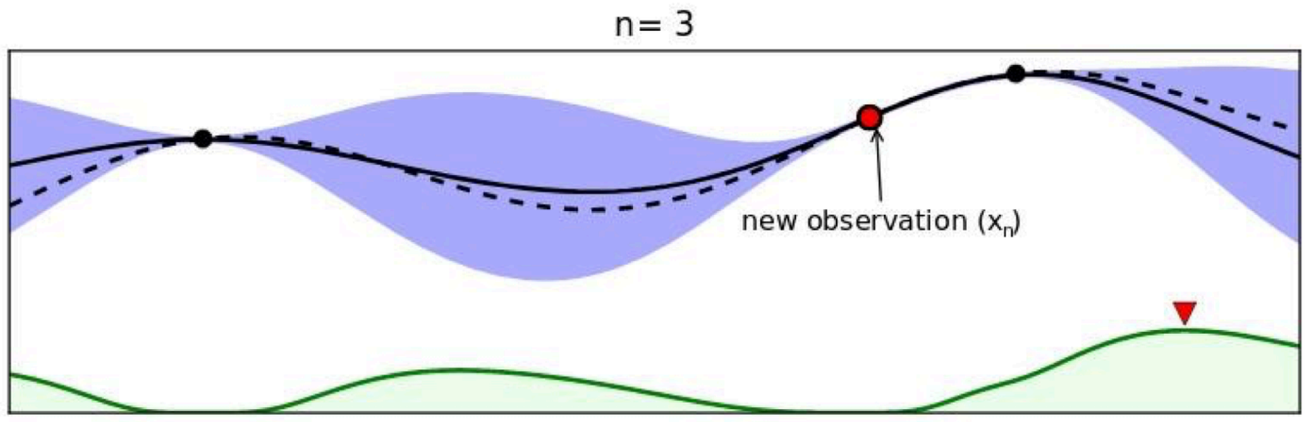
- We want to find the peak of our true function (e.g., accuracy as a function of hyperparameters)
- To find this peak, we will fit a Gaussian Process to our observed points and pick our next best point where we believe the maximum will be.
- This next point is determined by an accusation function which trades between the exploration and exploitation.

Lecture by Nando de Freitas and a Tutorial paper by Brochu et al

Bayesian Optimization



Find the next best point x_n that maximizes acquisition function



Evaluate f at the new observation x_n and update posterior

Update acquisition function from new posterior and find the next best point

Brochu et al., 2010, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning

Acquisition function

Guides the optimization by determining which point to observe next and is easier to optimize to find the next sample point.

The posterior GP gives the predictive mean and variance functions.

$$\gamma(x) = \frac{f(x_{best}) - \mu(x)}{\sigma(x)}$$

- Probability of Improvement (PI): (Kushner 1964)

$$a_{PI} = \Phi(\gamma(x))$$

- Expected Improvement (EI): (Mockus 1978)

$$a_{EI} = \sigma(x)[\gamma(x)\Phi(\gamma(x)) + \mathcal{N}(\gamma(x), 0, 1)]$$

- GP Upper/Lower Confidence Bound: (Srinivas 2010)

$$a_{LCB} = \mu(x) - \kappa\sigma(x)$$

Choices of GP

Ironic Problem!

Bayesian optimization has its own hyperparameters!

- Covariance function selection
- This turns out to be crucial to good performance.
- The default choice for regression is way too smooth.
  Instead: use adaptive Matern 3/5 kernel.
- Gaussian process hyperparameters
- Typical empirical Bayes approach can fail horribly.
  Instead: use Markov chain Monte Carlo integration.

GP Hyperparameters: MCMC to Rescue

- Marginalize over hyperparameters and compute integrated acquisition function.
- Covariance hyperparameters are often optimized rather than marginalized, typically in the name of convenience and efficiency.
- Compute accusation function by marginalizing over hyperparameters:

$$\hat{a}(x) = \int a(x;\theta)p(\theta|D_n)d\theta \approx \frac{1}{K}\sum a(x;\theta_k) \;\; where \; \theta_k \sim p(\theta|D_n)$$