Gaussian Processes and Active Learning

A. Gilad Kusne, aaron.kusne@nist.gov

Gaussian Process vs Neural Networks

- GPs are developed from theoretical foundations
  - Interpretability
- Theory based results
- Infinitely large (wide) NN -> GP (will come back to this)
- GPs are much slower -> Ongoing Research
- Start learning with simple methods and build up
  - Linear Regression, K-Nearest neighbors, k-means, etc.

Moving from linear model to GP

- Previous $f(\boldsymbol{x}) = \beta_0 + x_1\beta_1 + \cdots x_p\beta_p$

- We want a more general form, i.e. we want to explore a space of functions, and identify a f that best captures our data

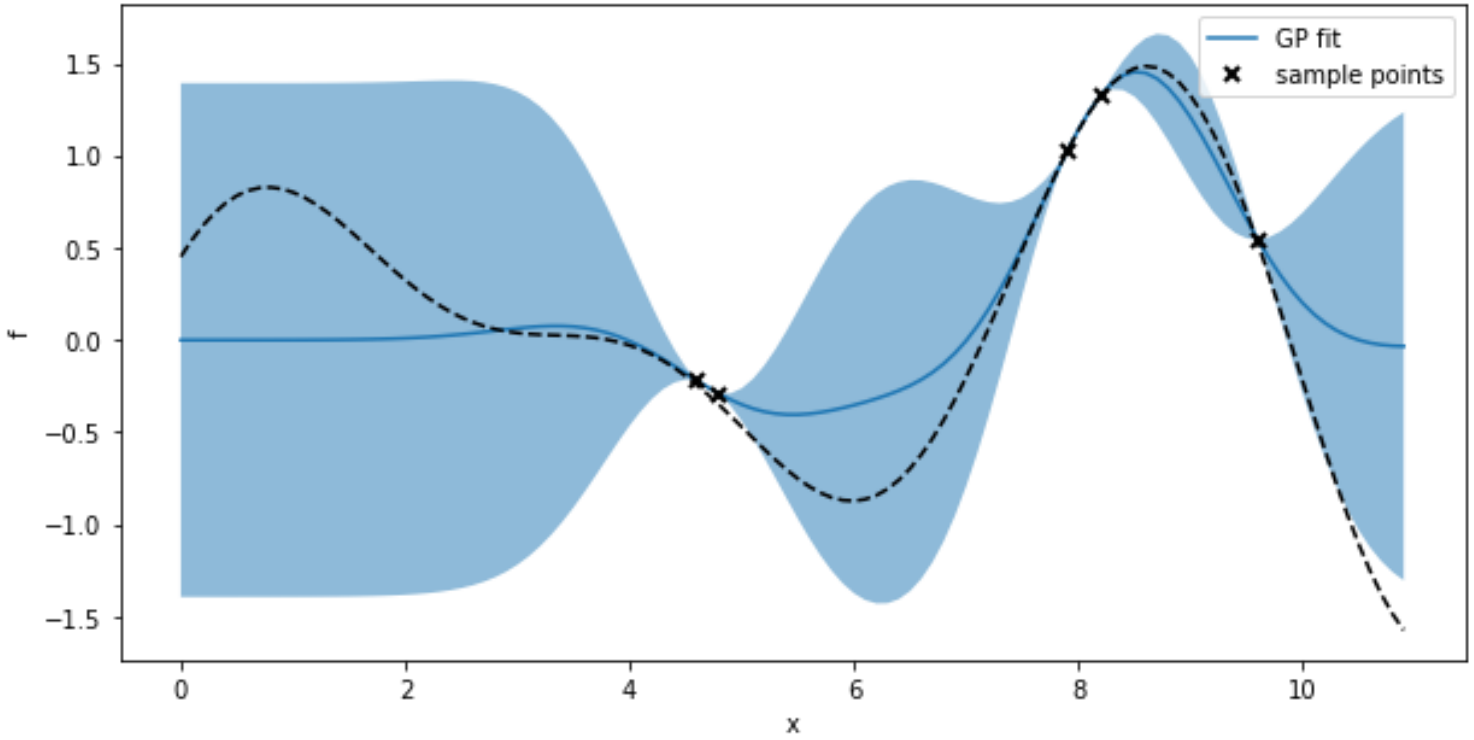
- A convenient space to search over is the multivariate Gaussian, i.e. GP's

$$f \sim N(\mathbf{0}, \boldsymbol{K})$$

- But what does this actually look like?

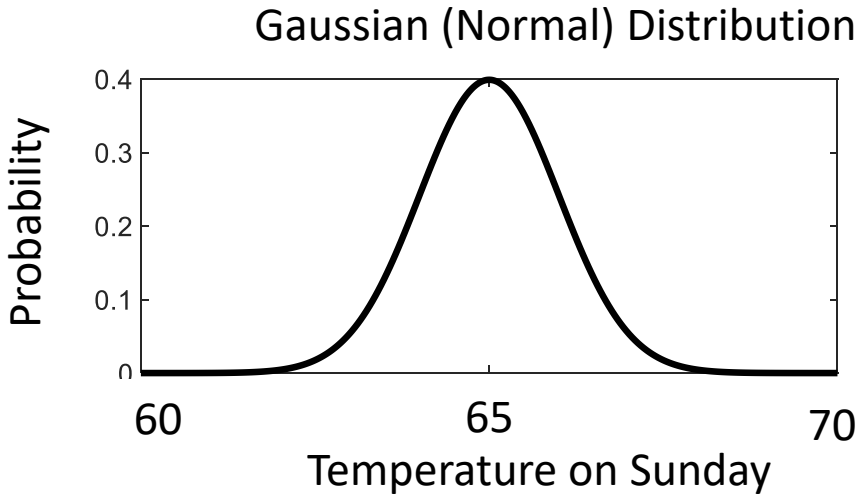
Gaussian Process

- Inputs: Data points
- Outputs: Mean f(x); Variance: s(x)



1 Random Variable

- Normal distribution, e.g. X ~ N(65, 3)

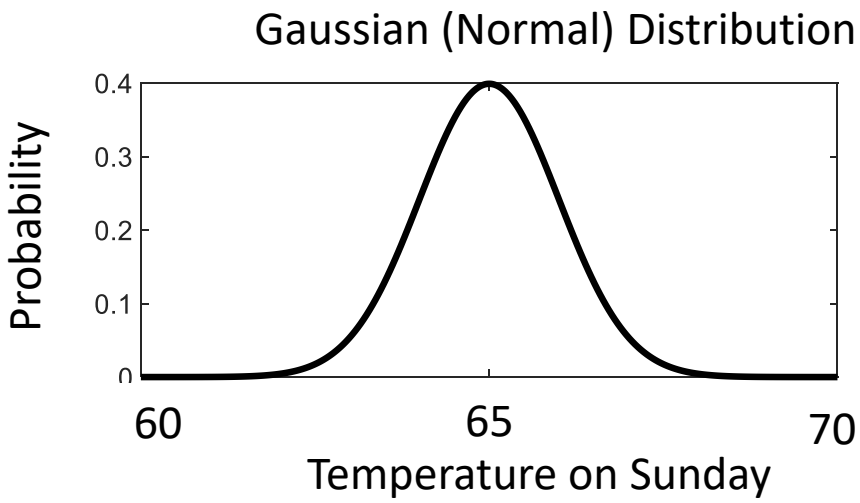


Gaussian Process

- Random variable e.g. X ~ N(65,1)
- Random Process $X = \{X_1, X_2, X_3, \dots\}$
  - Collection of random variables that are indexed.
- Gaussian Process = Random Process based on Gaussian:

$$P(X) \sim N(m, K)$$

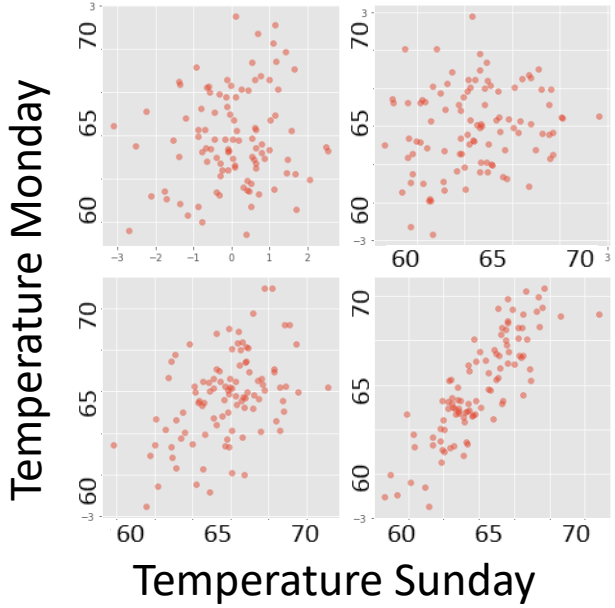
- The variables are related by covariance matrix K (also called the Kernel matrix)

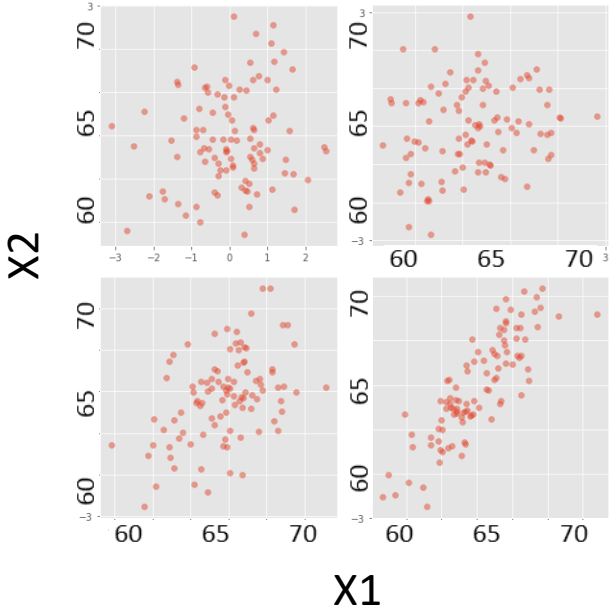


Gaussian Processes: From sampling a multivariate Gaussian to function space

- Each subplot corresponds to a different correlation matrix
  - As correlation increases variation in function space decreases

- Takeaway: every sample drawn from our Gaussian is a curve/function

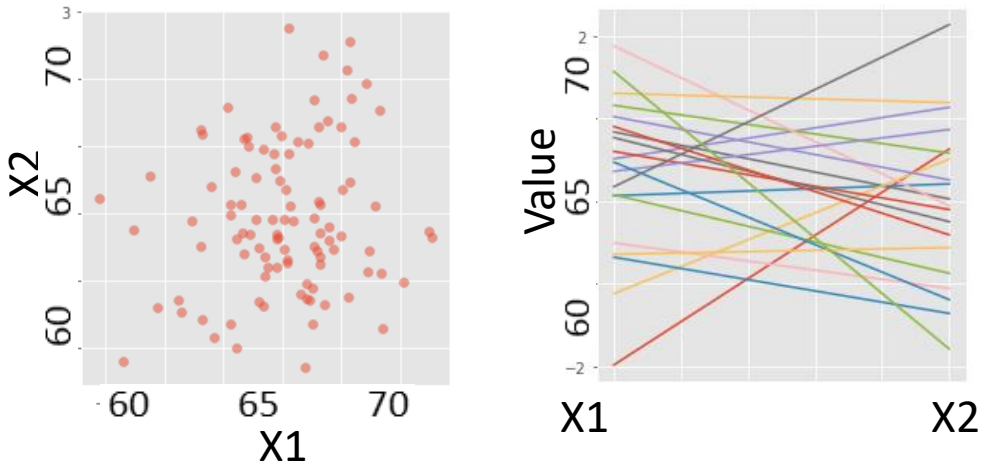




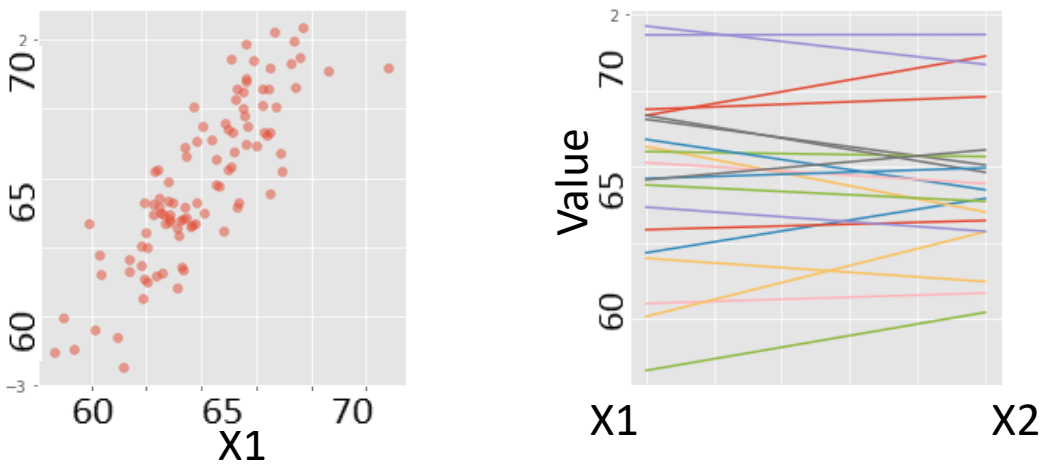
Gaussian Processes: From sampling a multivariate Gaussian to function space

- Each subplot corresponds to a different correlation matrix
  - As correlation increases variation in function space decreases
  - Think of this as more/less smoothing
- Takeaway: every sample drawn from our Gaussian is a curve/function

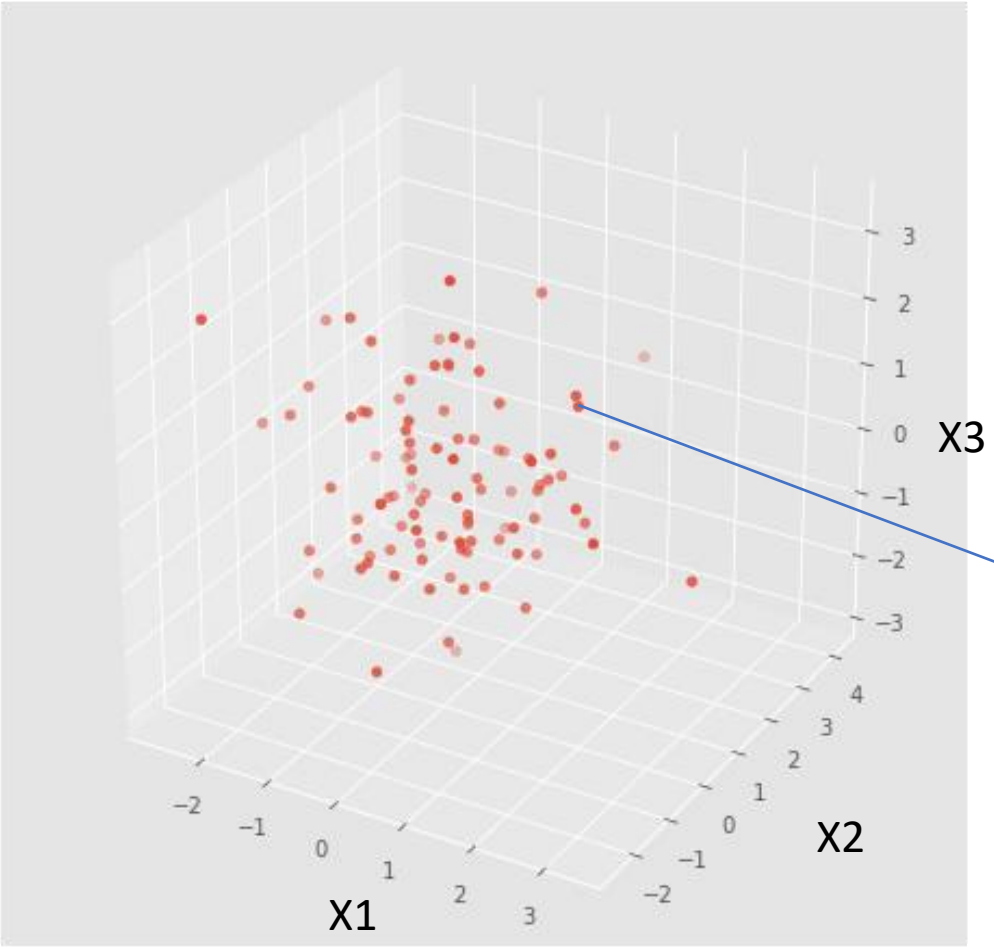
Uncorrelated

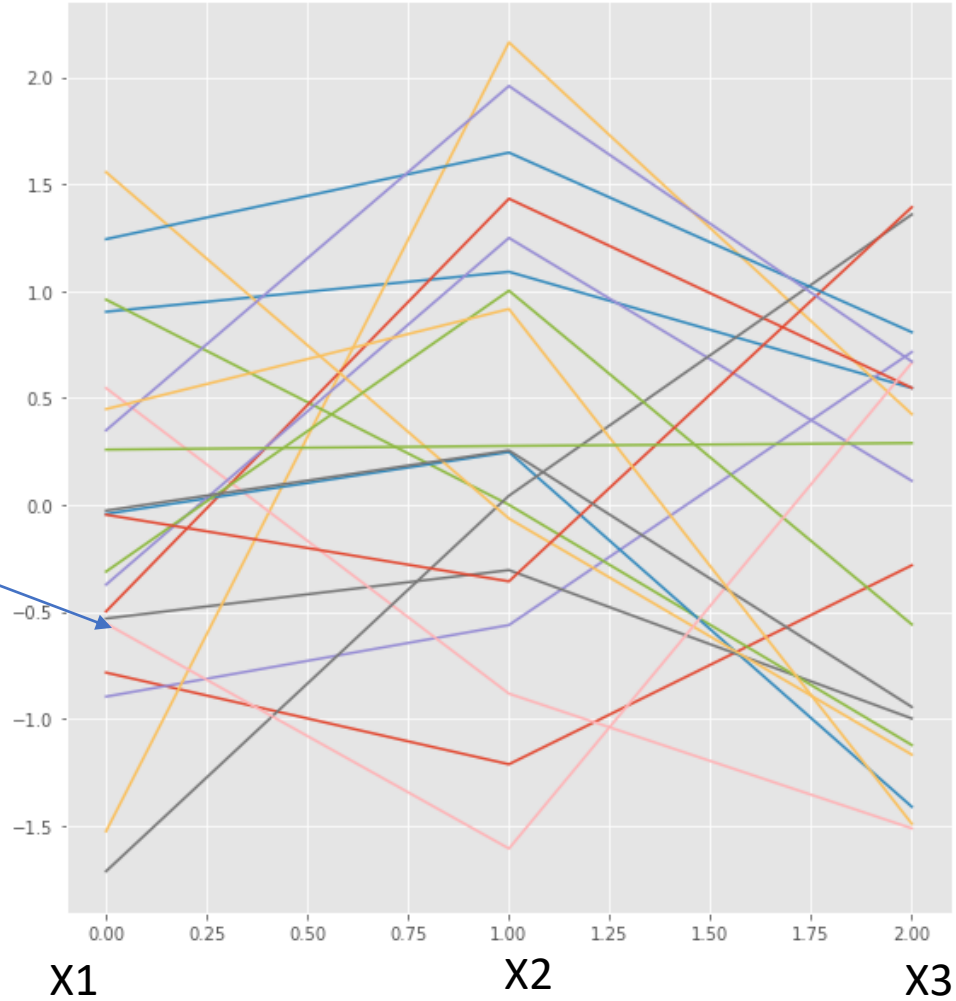


Correlated

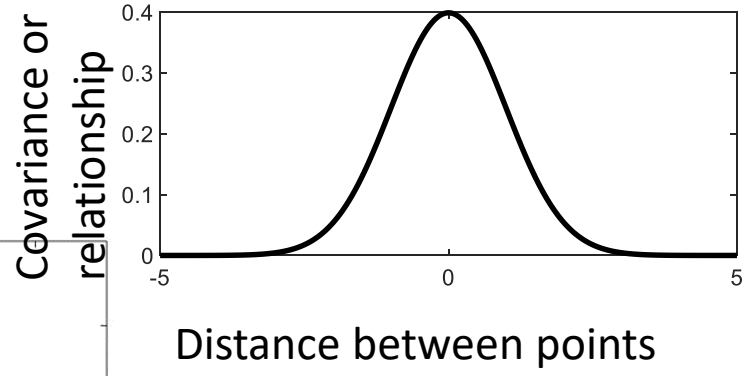


Another example using 3 Variables

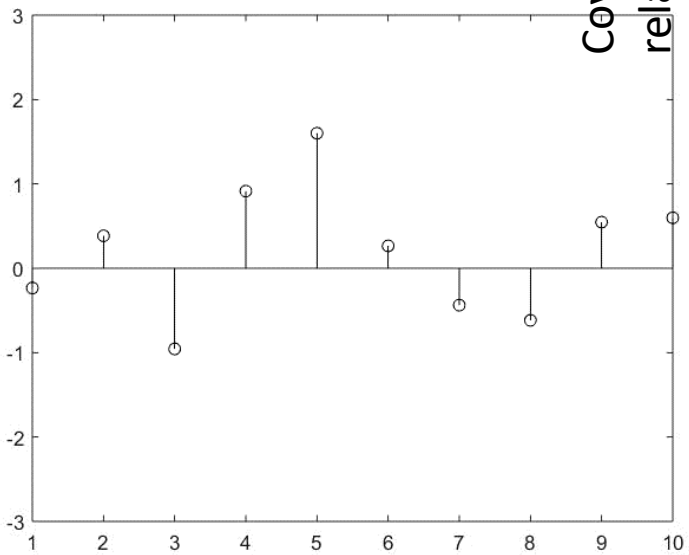




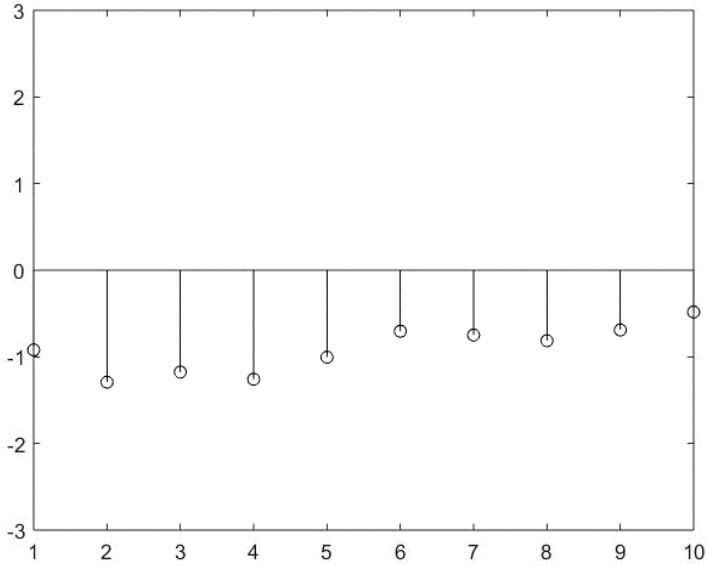
Gaussian Process



- Random process with mean = 0 (Independent variables)



- Gaussian Process with mean = 0, covariance $k(x, x') = \alpha \exp\left(-\frac{\|x-x'\|_2^2}{2\gamma^2}\right)$



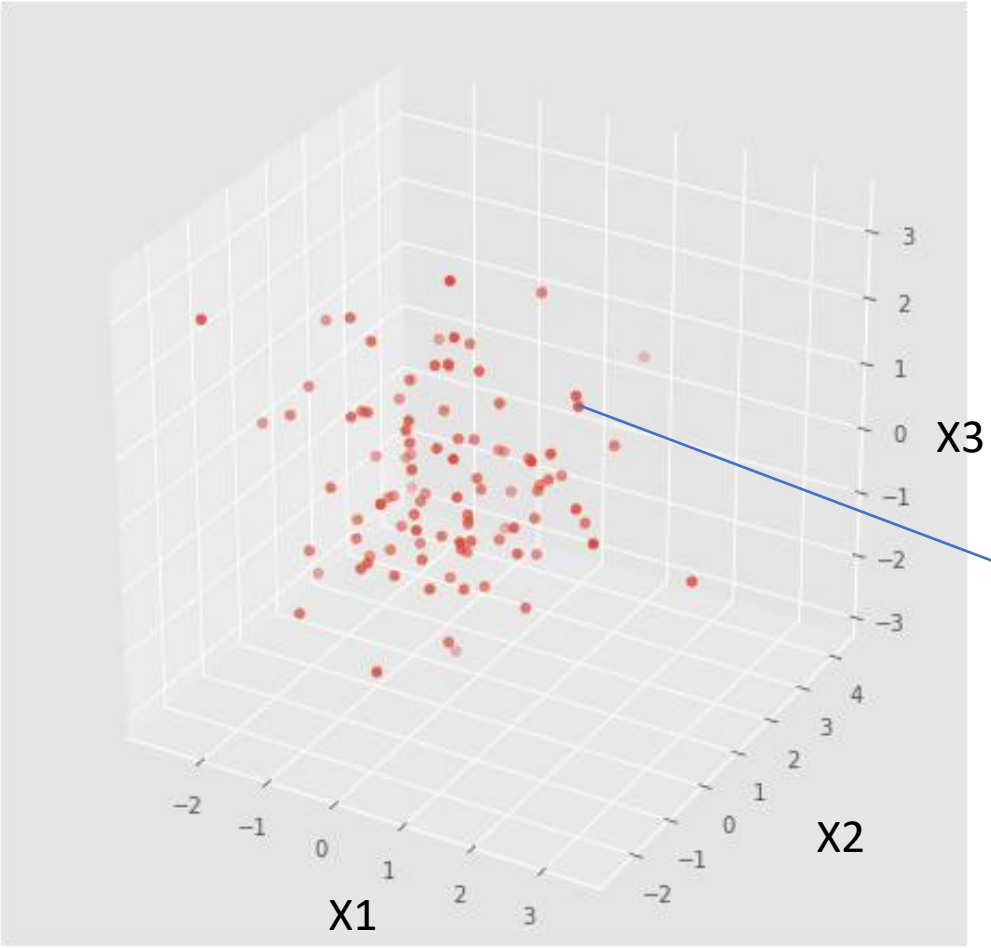
Covariance

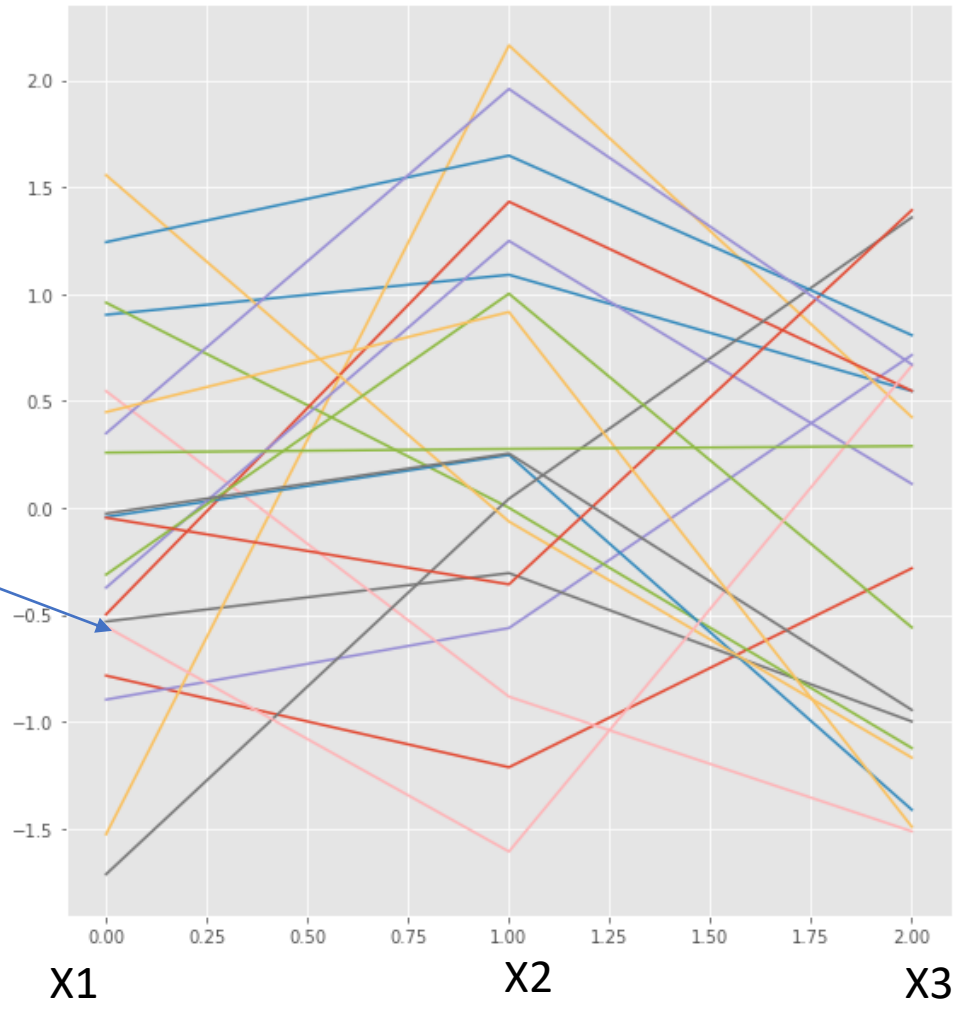
- $\mathrm{cov}(y_1, y_2) = \frac{1}{n}\sum_{i=1}^{n}\left(y_{1,i} - E(y_1)\right)\left(y_{2,i} - E(y_2)\right) = \sigma_{1,2}$

- $K = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \cdots & \sigma_{n,n} \end{bmatrix}$

Variance

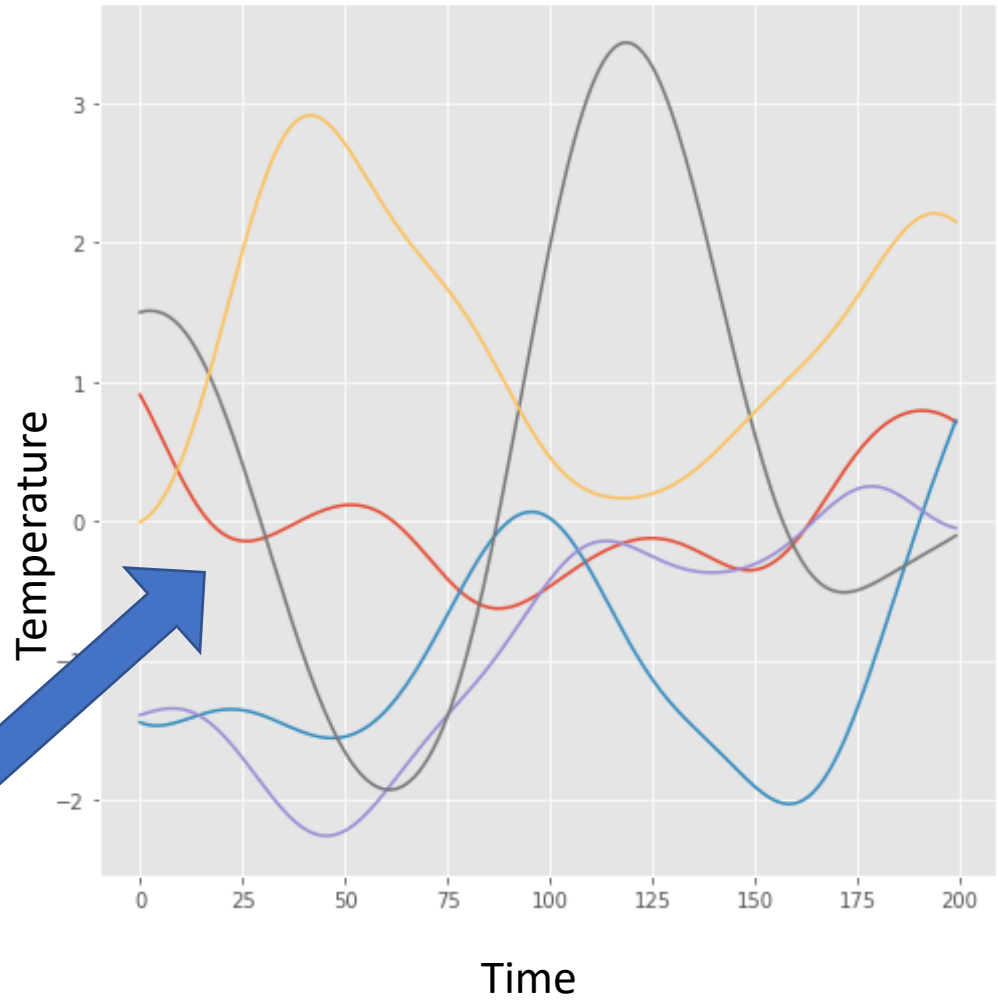
Another example using 3d Gaussian





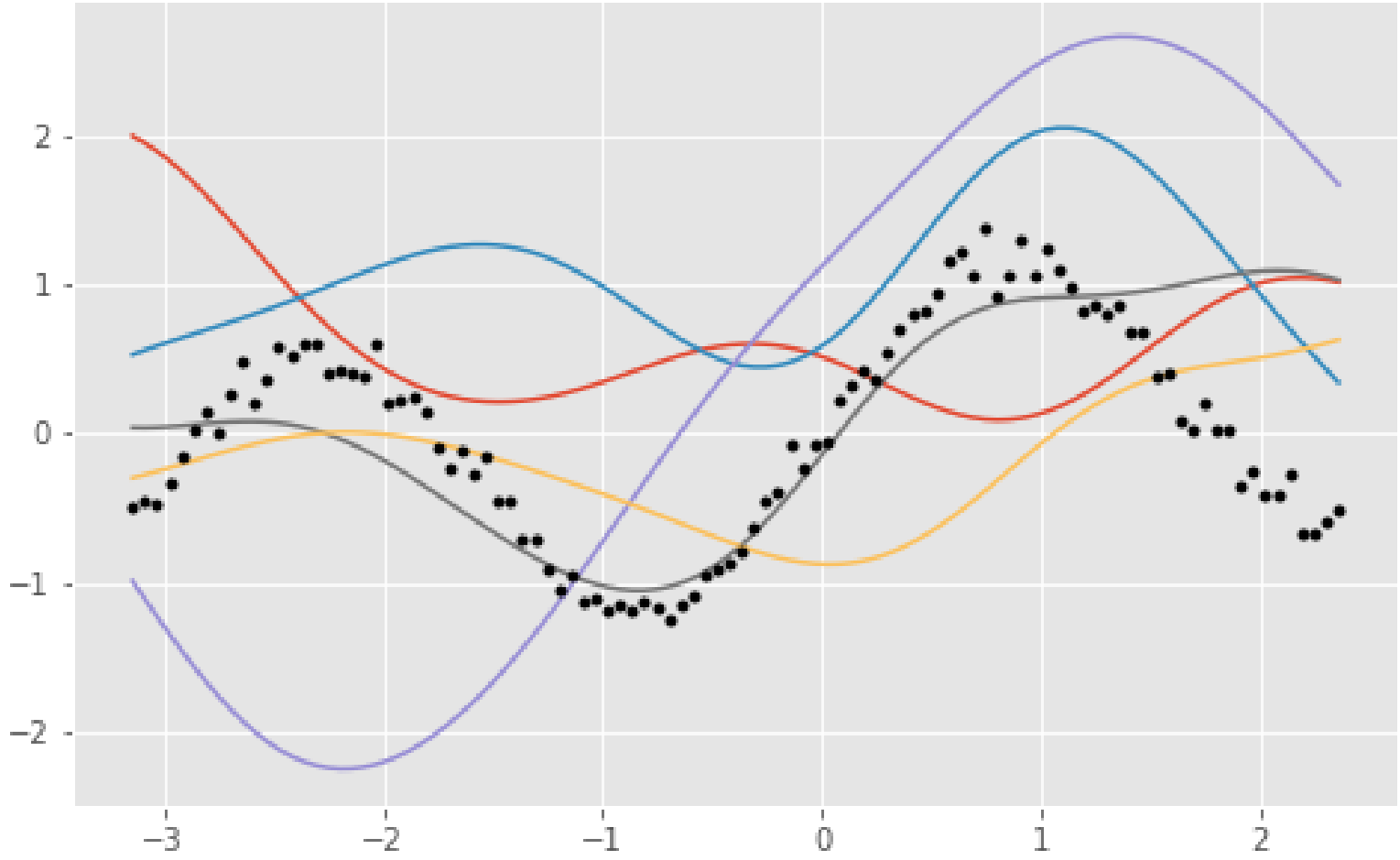
A space of prior functions/curves

- Extend to infinite variables.
- Each position on the x axis corresponds to a variable.
- The corresponding value is given by the y-value.
- We now have curves! $f(\boldsymbol{x}) \sim N(0, \boldsymbol{K})$
- Correlation is visible through curve smoothness.



Each curve is a Sample!

Regression



How do we go from random functions to something that reflects the behavior of our data?

Gaussian Process: Bayes Rule

- Goal:
  - Inputs: Data *D*
  - Outputs: Mean f(x); Variance: s(x)

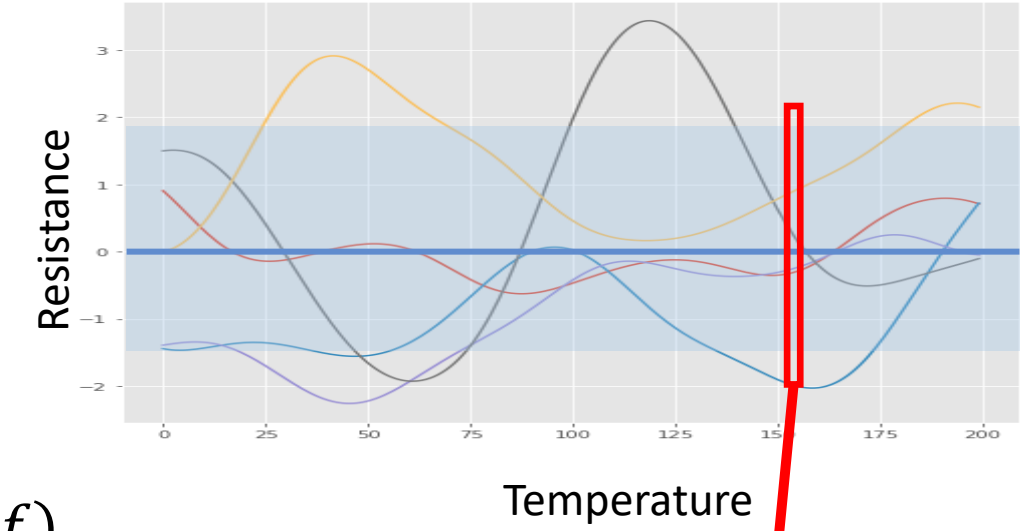
- Based on Bayesian Rule (posterior):

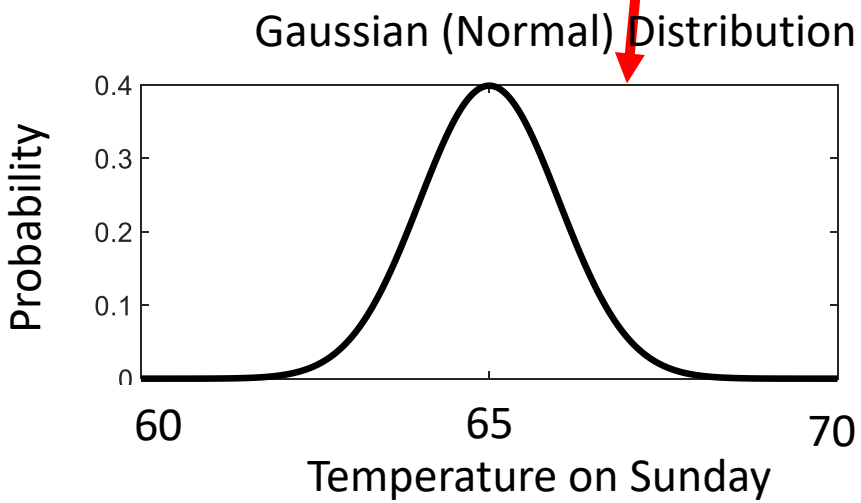
$$P(f|D) = \frac{P(D|f)P(f)}{P(D)}$$

- And Gaussian (Normal) Distribution:

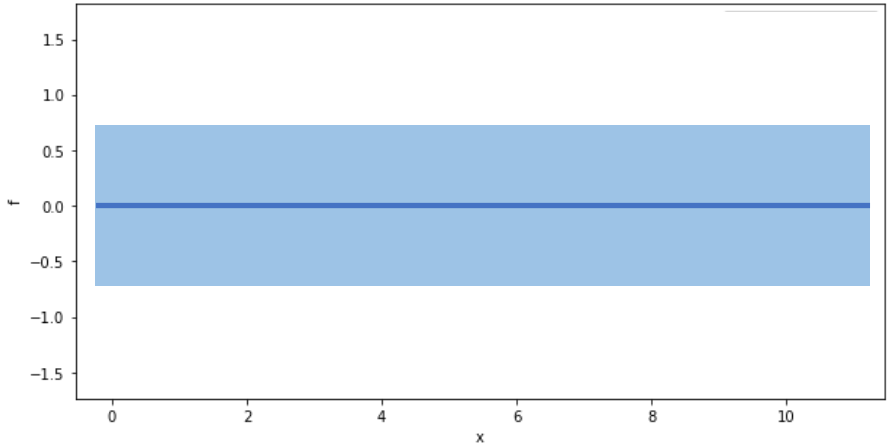
$$P(f) \sim N(0, K)$$
$$P(y|f, \sigma^2) \sim N(f, \sigma^2 I)$$

- Normal Posterior = Normal Likelihood * Normal Prior





Gaussian Process: Bayes Rule



- Inputs: Data *D*
- Outputs: Mean f(x); Variance: s(x)

- Based on Bayesian Rule (posterior):

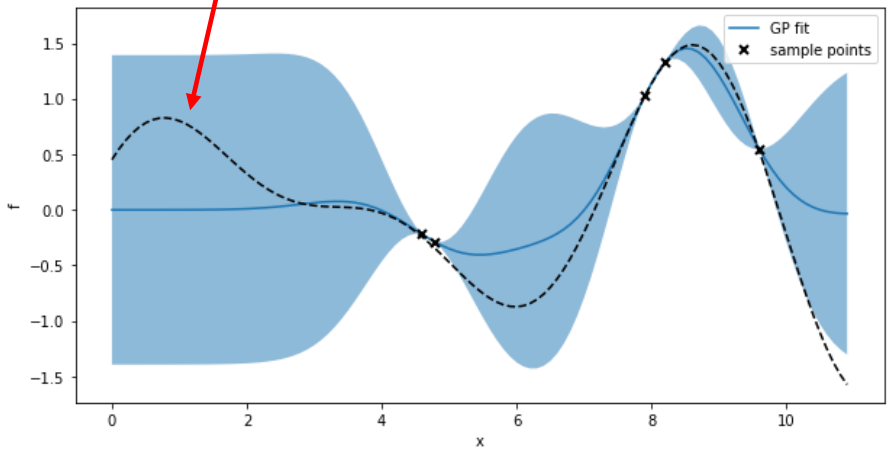
$$P(f|D) = \frac{P(D|f)P(f)}{P(D)}$$

- And Gaussian (Normal) Distribution:

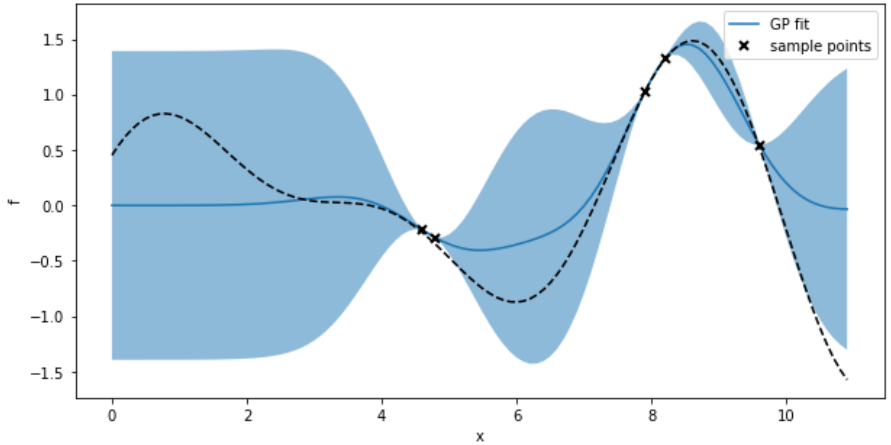
$$P(f) \sim N(0, K)$$
$$P(y|f, \sigma^2) \sim N(f, \sigma^2 I)$$

- Normal Posterior = Normal Likelihood * Normal Prior

True, Generating Function



Gaussian Process: Quick Review



- Inputs: Data *D*
- Outputs: Mean f(x); Variance: s(x)

- Based on Bayesian Rule (posterior):

$$P(f|D) = \frac{P(D|f)P(f)}{P(D)}$$

- Mean and uncertainty with each new data point

$$\mu(x^*) = \mathbf{K}_{*x}(\mathbf{K}_{xx} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$
$$\sigma^2(x^*) = \mathbf{K}_{**} - \mathbf{K}_{*x}(\mathbf{K}_{xx} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{*x}^T$$
$$N(\mu, K)$$

Kernel Length Scale & Noise Variance

- These parameters can be estimated directly from the log likelihood

$$\max_{\sigma,\gamma} \log\big(P(y|\sigma,\gamma)\big) = \max_{\sigma,\gamma}[-\frac{1}{2}\boldsymbol{f}^T\boldsymbol{K}_\gamma\boldsymbol{f} - \frac{1}{2}\log\big(\det\big(\boldsymbol{K}_\gamma + \sigma^2\boldsymbol{I}\big)\big)$$

- Recalling that $k(x,x') = \exp\left(-\frac{\|x-x'\|_2^2}{2\gamma^2}\right)$

Connecting back to linear regression

- Recall that in our linear model $f(x) = x_1\beta_1 + x_2\beta_2$

- To introduce non-linearity we could (for example) use polynomial terms

$$f(x) = x_1\beta_1 + x_2\beta_2 + x_1^2\beta_{11} + x_2^2\beta_{22} + x_1x_2\beta_{12}$$

- Can be written as $f(x) = \phi(\boldsymbol{x})\boldsymbol{\beta}$
  - $\phi(\boldsymbol{x}) = (x_1, x_2, x_1^2, x_2^2, x_1x_2)$ is a "polynomial" basis
  - $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_{11}, \beta_{22}, \beta_{12})$

- We pick ϕ to capture different types of behavior and potential non-linearities in our data

Choice of basis

- The choice of $\boldsymbol{\phi}$ can be quite general - *does not need to be explicitly defined*
  - just the existence of it's inner product (e.g. $k(x, x') = \exp\left(-\frac{\|x-x'\|_2^2}{2\gamma^2}\right)$)

- NB: $\boldsymbol{K} = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T$ is a $n \times n$ matrix of inner products

$$\boldsymbol{K} = \begin{pmatrix} \phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_1) & \phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_1) & \cdots & \phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_n) \\ \phi(\boldsymbol{x}_2)^T\phi(\boldsymbol{x}_1) & \phi(\boldsymbol{x}_2)^T\phi(\boldsymbol{x}_2) & \cdots & \phi(\boldsymbol{x}_2)^T\phi(\boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_1) & \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_2) & \cdots & \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_n) \end{pmatrix}$$

- Kernel Trick: only need to know the form of the inner product not the mapping $\phi(\cdot)$ (plus a few other conditions

Connecting back to Ridge Regression

- Recall solution to ridge regression: $\boldsymbol{\beta} = (\boldsymbol{\phi}^T\boldsymbol{\phi} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{\phi}^T\boldsymbol{y}$
  - If we drop $\gamma\boldsymbol{I}$ this is just the linear regression solution

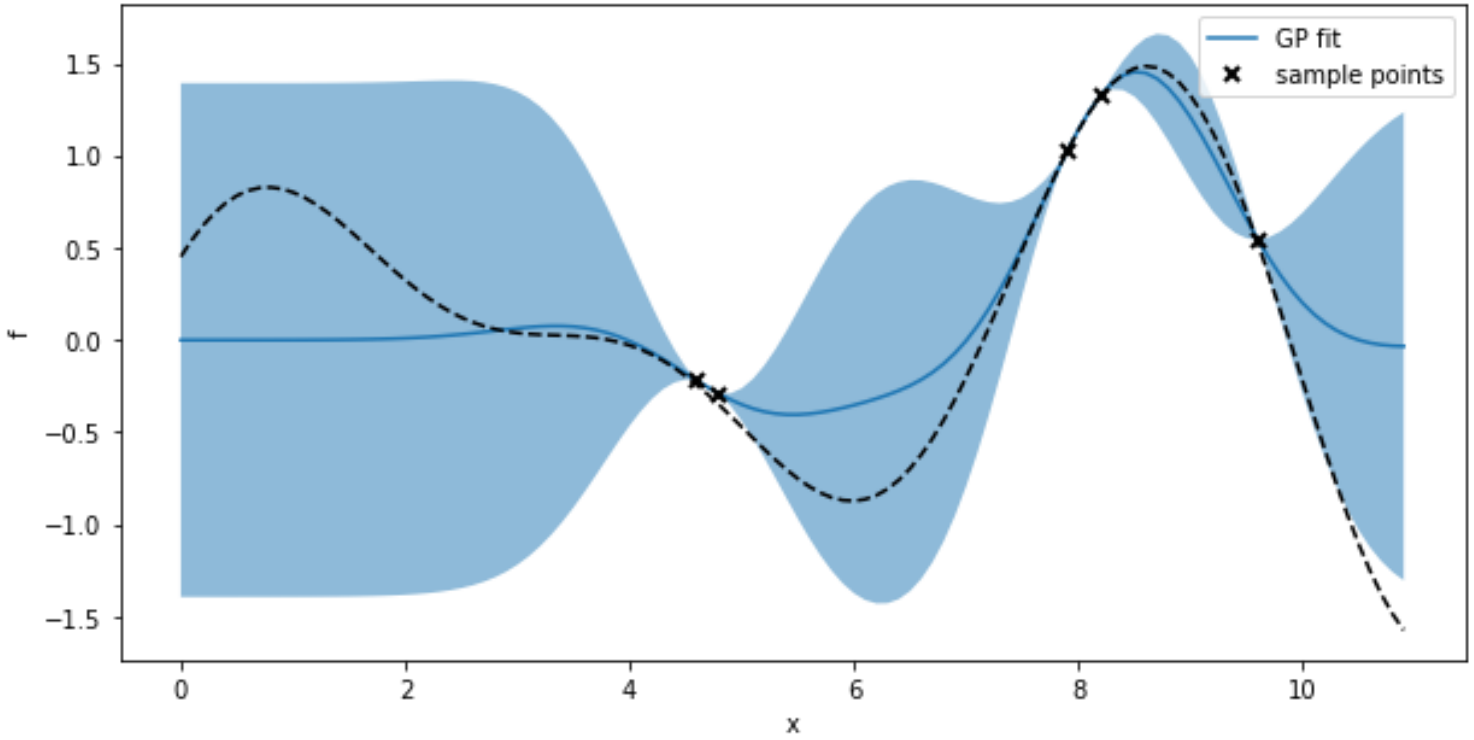
- Using matrix inversion lemma:

$$(\boldsymbol{\phi}^T\boldsymbol{\phi} + \gamma\boldsymbol{I})^{-1}\boldsymbol{\phi}^T = \boldsymbol{\phi}^T(\boldsymbol{\phi}\boldsymbol{\phi}^T + \sigma^2\boldsymbol{I})^{-1}$$

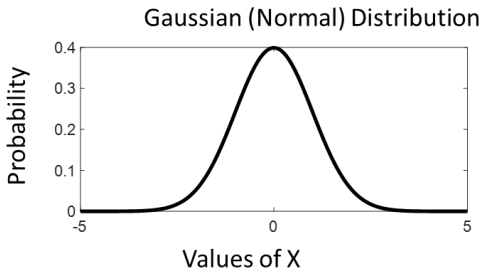
- With $\widehat{\boldsymbol{y}} = \boldsymbol{\phi}\boldsymbol{\beta} = \boldsymbol{\phi}\boldsymbol{\phi}^T(\boldsymbol{\phi}\boldsymbol{\phi}^T + \gamma\boldsymbol{I})^{-1}\boldsymbol{y} = \boldsymbol{K}(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$
  - This is exactly the posterior mean in our GP
  - NB: same solution can be arrived at by noting $\boldsymbol{\beta} = \boldsymbol{\phi}^T\boldsymbol{\alpha}$ and solving for $\boldsymbol{\alpha}$

Gaussian Process

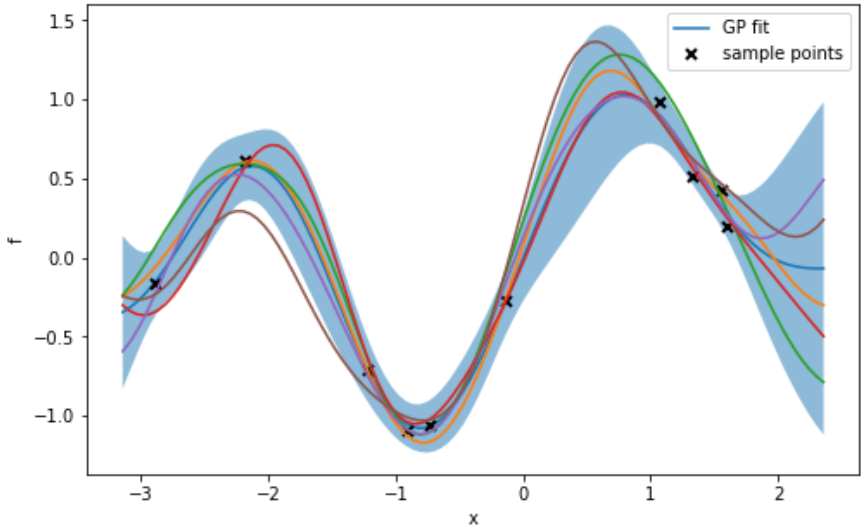
- Inputs: Data points
- Outputs: Mean f(x); Variance: s(x)

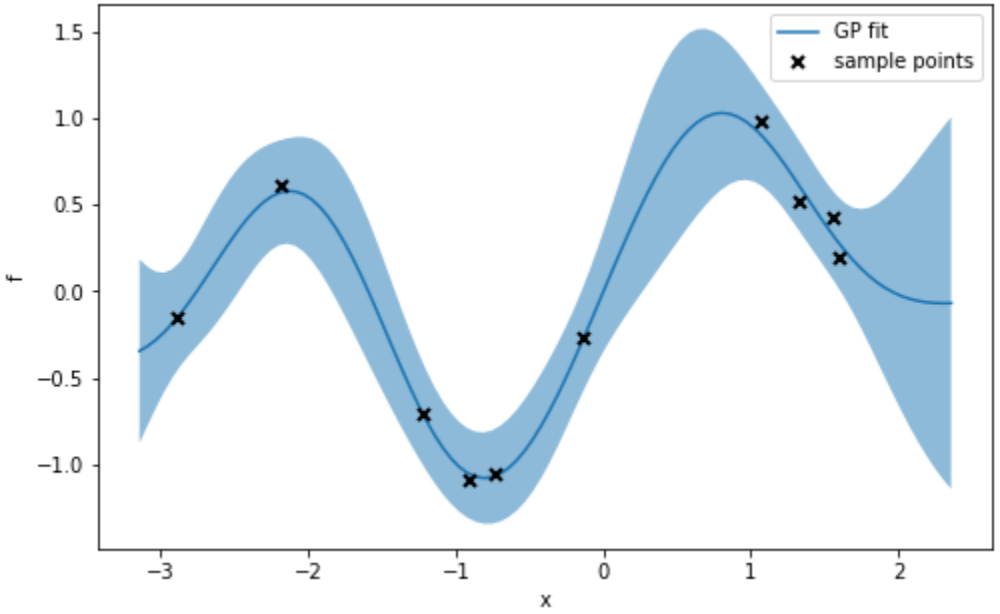


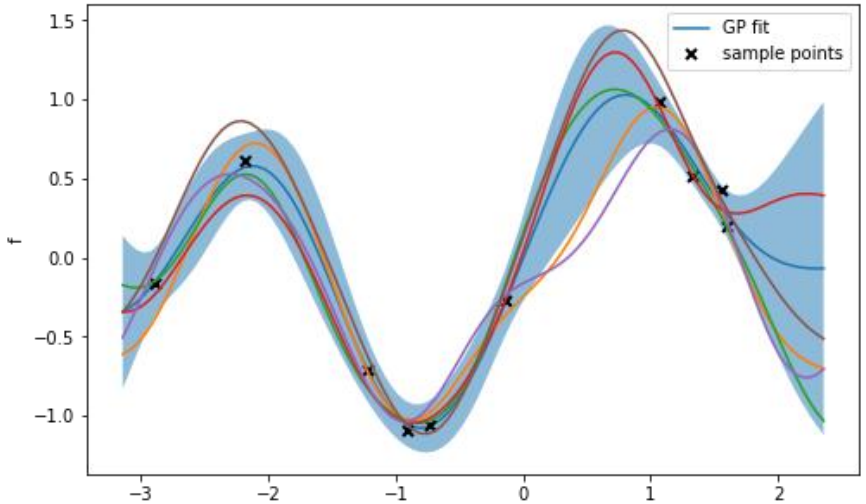
Gaussian Process: Quick Review



- Sampling the distribution
- Sample stochastic variable ->
- Sample stochastic process

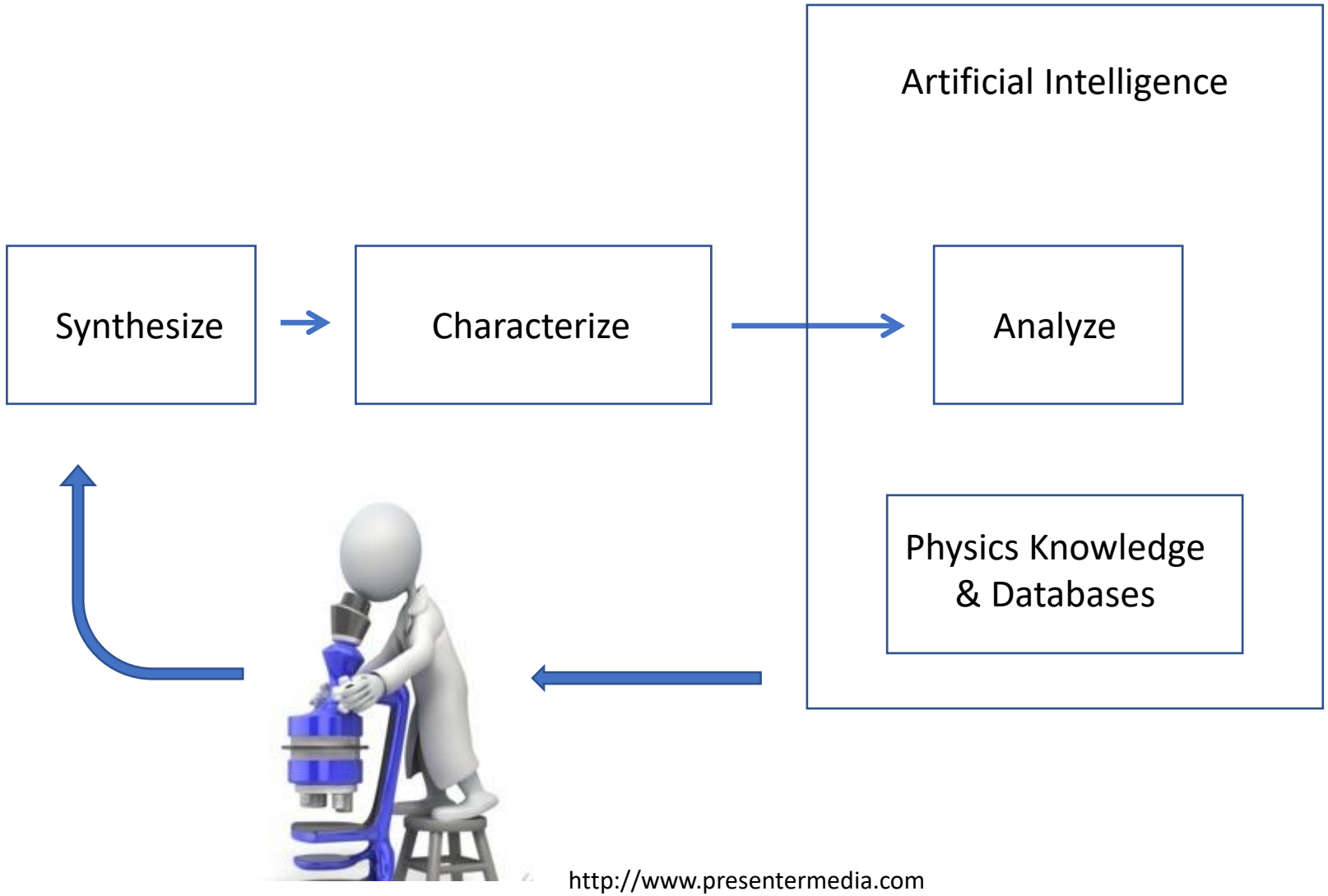






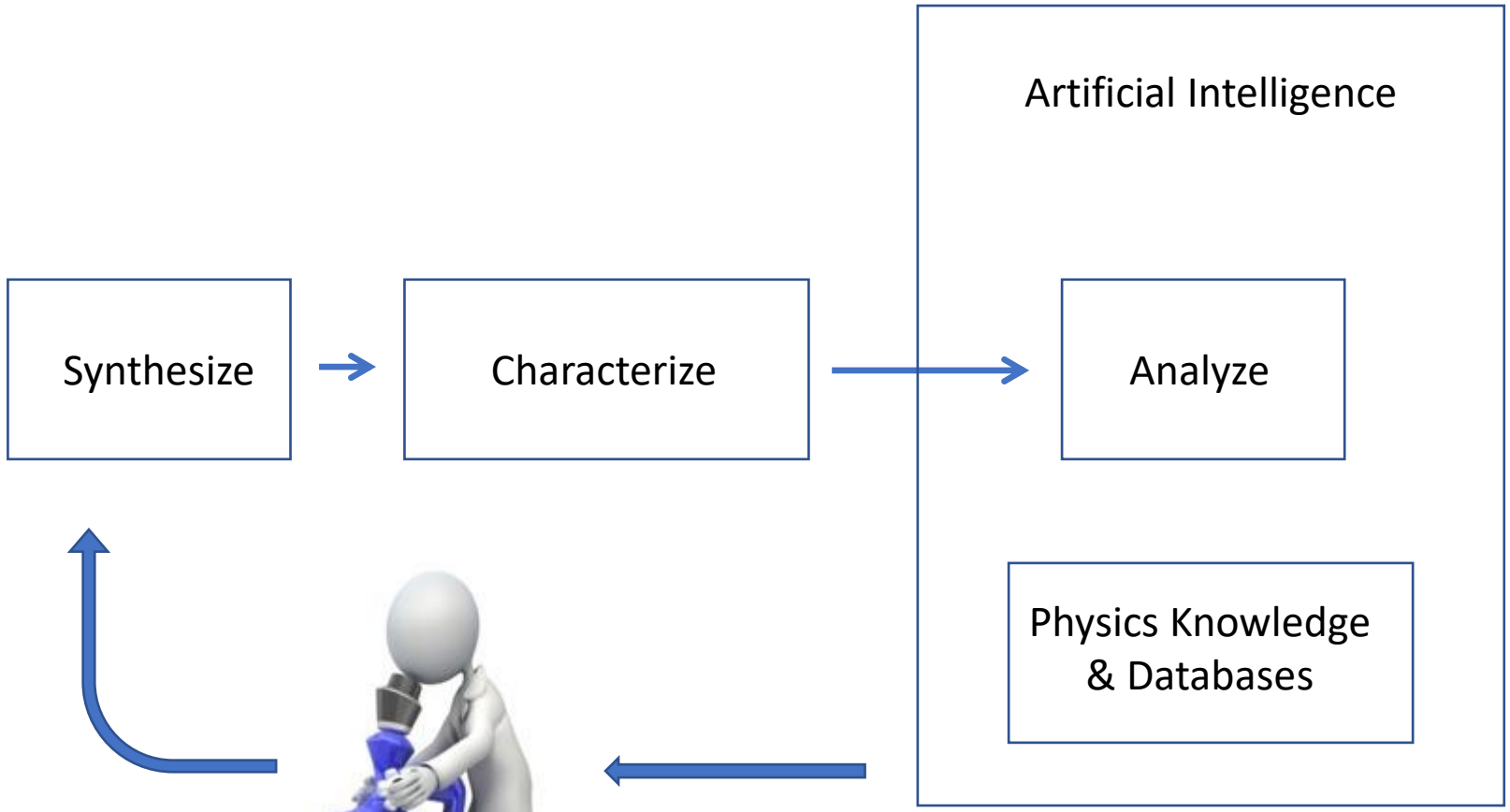
Levels of AI interaction

Machine Learning Informed



http://www.presentermedia.com

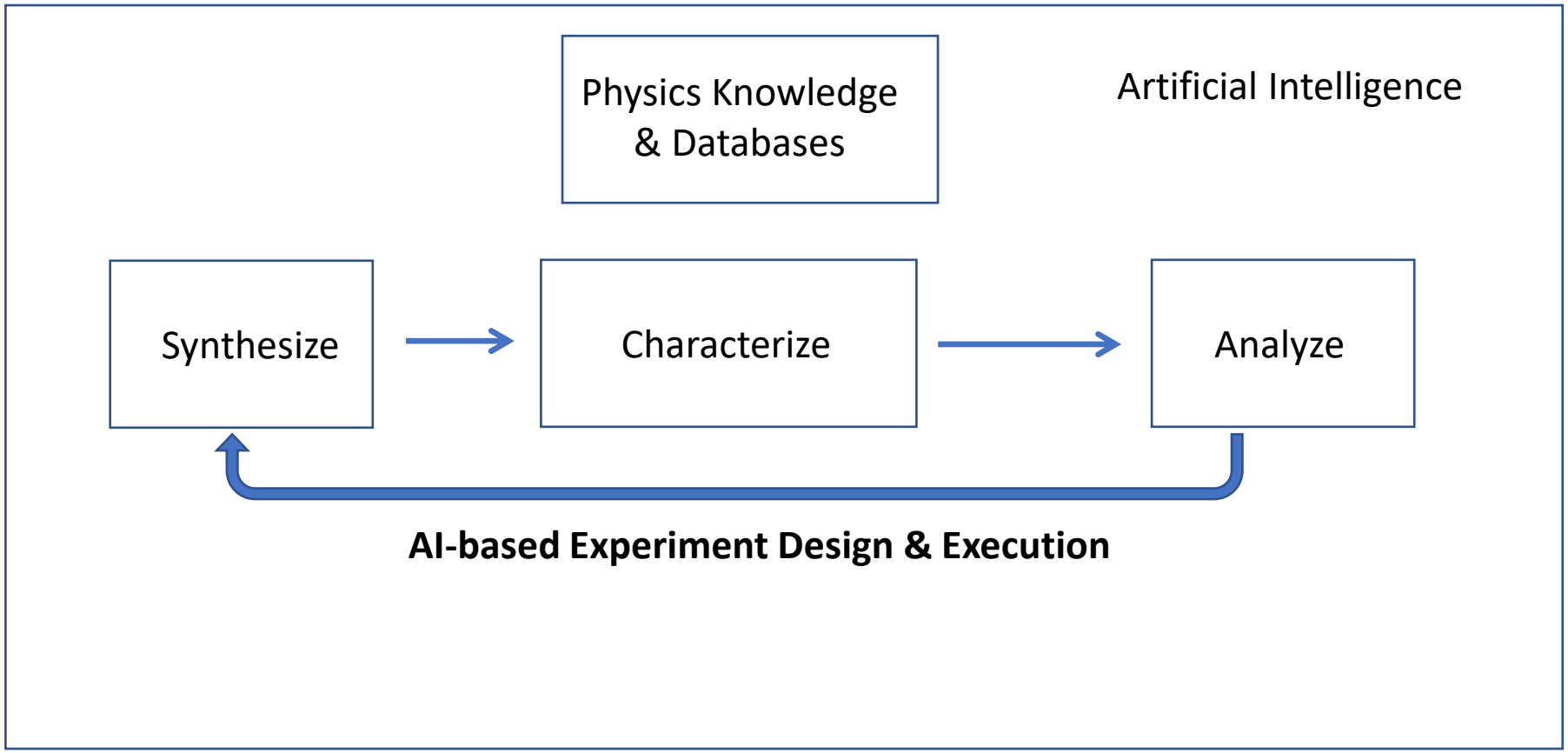
Active Learning Driven



ML says: “Do this experiment!”

http://www.presentermedia.com

Autonomous Materials Research

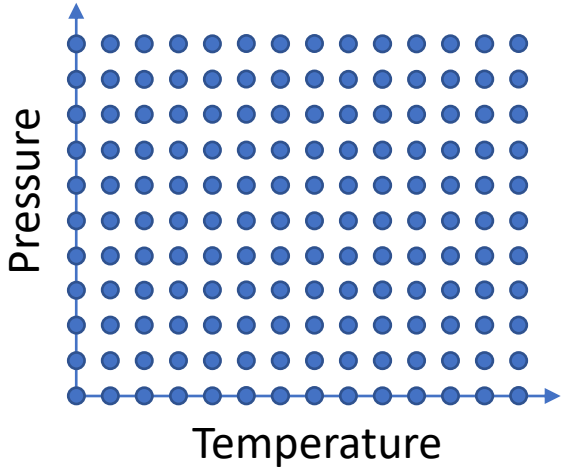


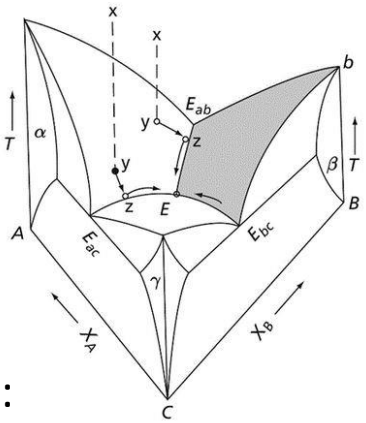
The Challenge of Materials Exploration

Complex materials described by High dimensional space!

Exhaustive Search:

100 experiments over values of A





4 parameters:
3 Elements + Temperature

Assume: For each parameter, 100 experiments over range.

$(10^2)^2 = 10^4$ experiments

4 Parameters -> 10^8 experiments

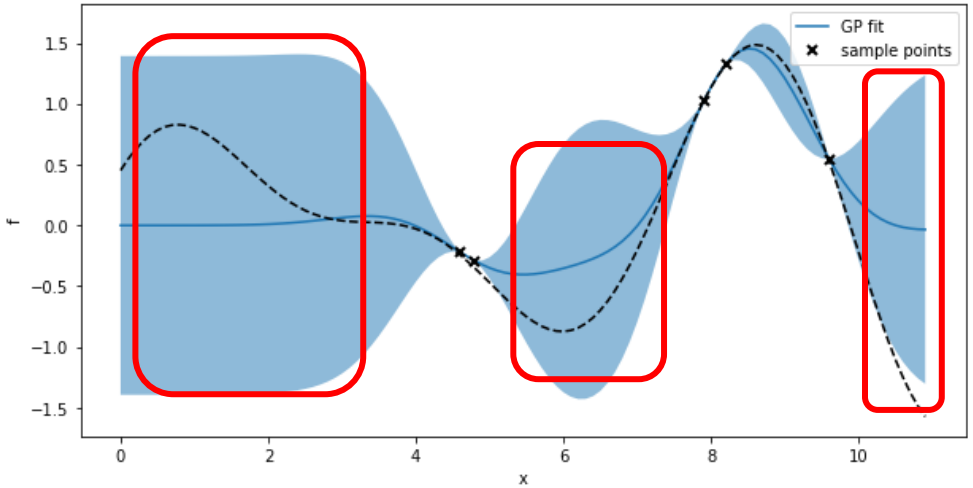
For N parameters -> $10^{2(N)}$ experiments!

Complex materials and complex materials physics are out of reach!

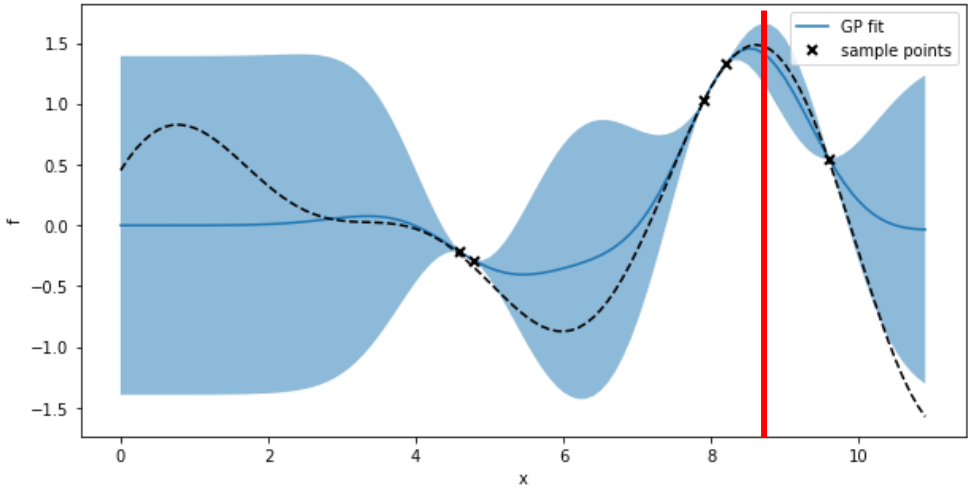
Active Learning

- *Actively* choose new data points to *learn* something about the search space
- 2 Different goals:
- Global mapping: know the value of the function everywhere.
  - Also (somewhat unhelpfully) known as: Active Learning
- Global optimum: Quickly find the best (i.e. max or min) value.
  - $x^* = \underset{x}{\operatorname{argmax}}\, f(x)$
  - Also called Bayesian Optimization or just BayesOpt

Global Mapping



Global Optimum



Active Learning

What do we need?

- $f(x)$ to represent our data as it comes in.
  - For Bayesian methods $f(x)$ must provide estimate and uncertainty
  - e.g. $\{\hat{f}(x),\ \sigma^2(x)\}$ ← Use GP!

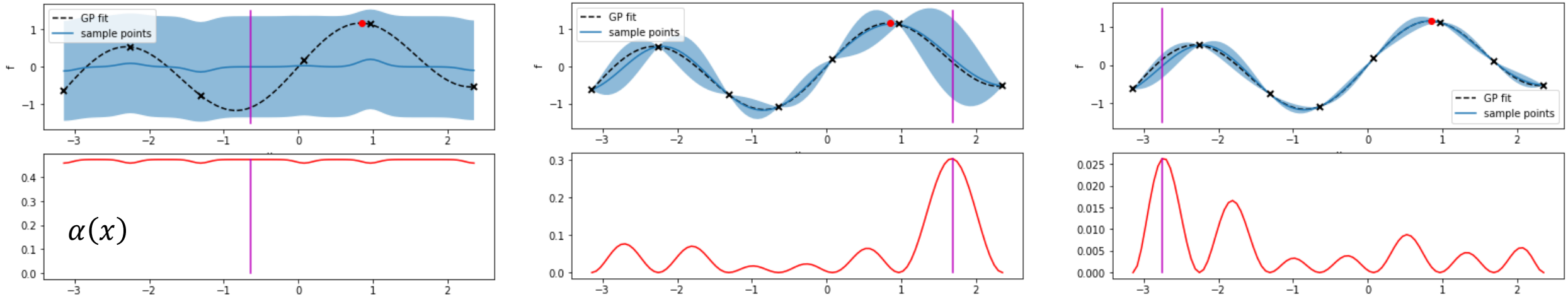
- $\alpha(x)$ the acquisition function which quantifies the desirability of performing each experiment.

This is Active Learning -upper category, active learning -lower category, & BO

Exploring

- Goal: Map the entire space.
- Method 2: Use the GP variance = Pure exploration

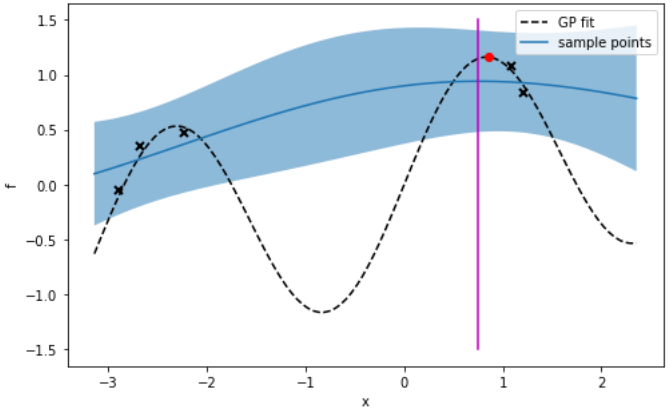
- $\alpha(x) = \sigma_{GP}^2(x)$
- $x^* = argmax\left(\sigma_{GP}^2(x)\right)$

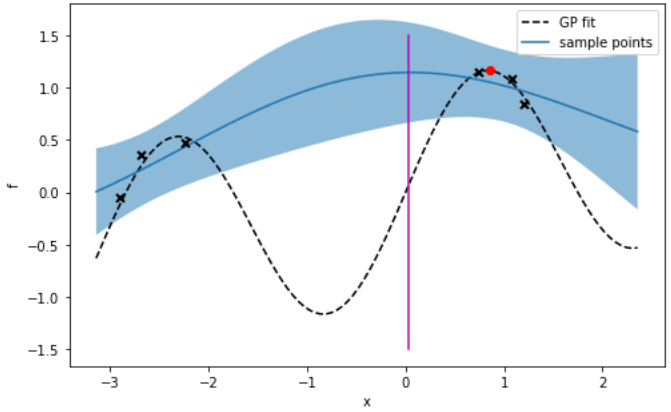


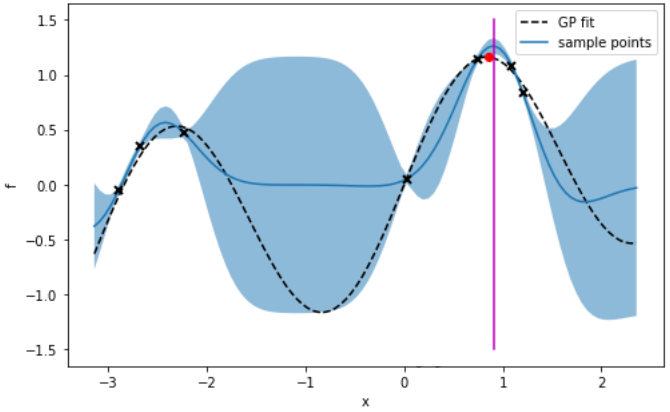
Bayesian Optimization: Exploiting Prior Knowledge

- Goal: Find a maximum or minimum
- Method 1: Use the GP mean = pure exploiting
- Trapped in local optima

- $\alpha(x) = \mu_{GP}(x)$ <- acquisition function
- $x^* = argmax(\mu_{GP}(x))$







BO: Upper Confidence Bounds

Explore

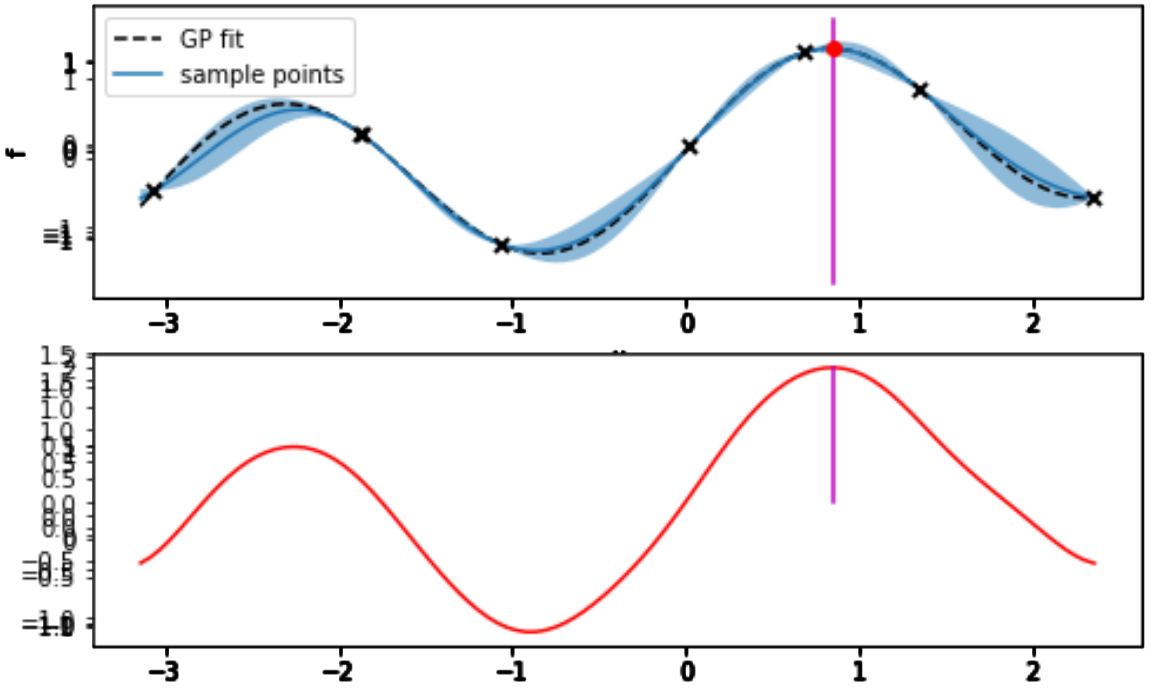
- $\alpha(x) = \mu_{GP}(x) + \beta\sigma_{GP}^2$

Exploit

- $x^* = argmax(\alpha(x))$

- β can be scheduled
- Example: $\beta = \sqrt{|D|n^2\pi^2/(6\lambda)}$
- As n increases, explore more

If β is 1? Maximum of top of envelope



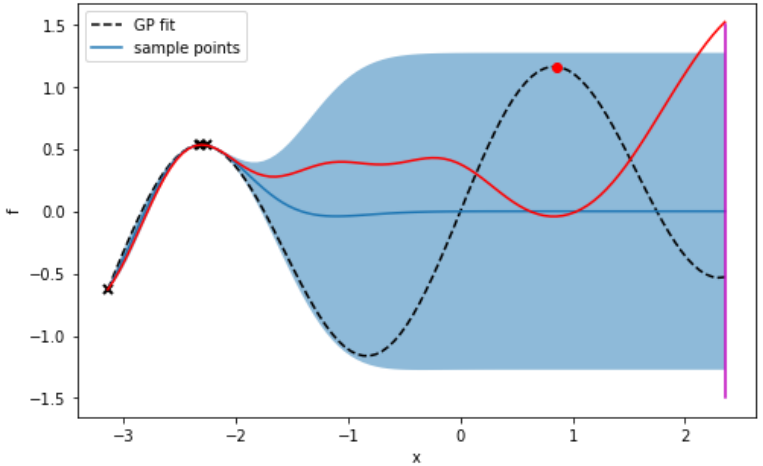
Thompson Sampling

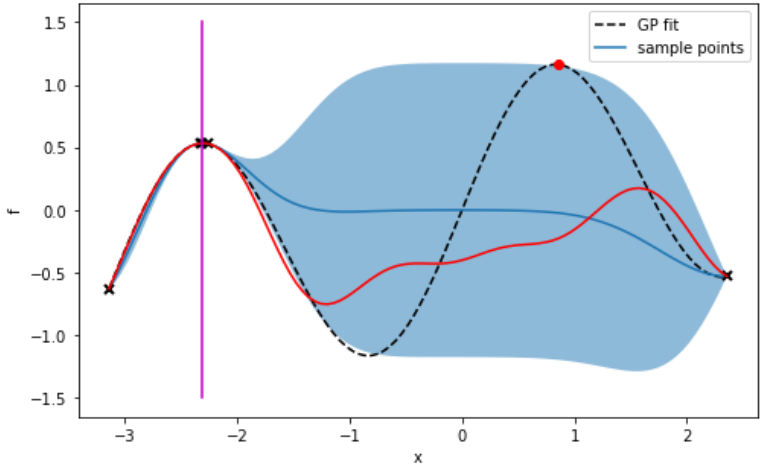
- Sample: $\tilde{f} \sim N(m, K)$

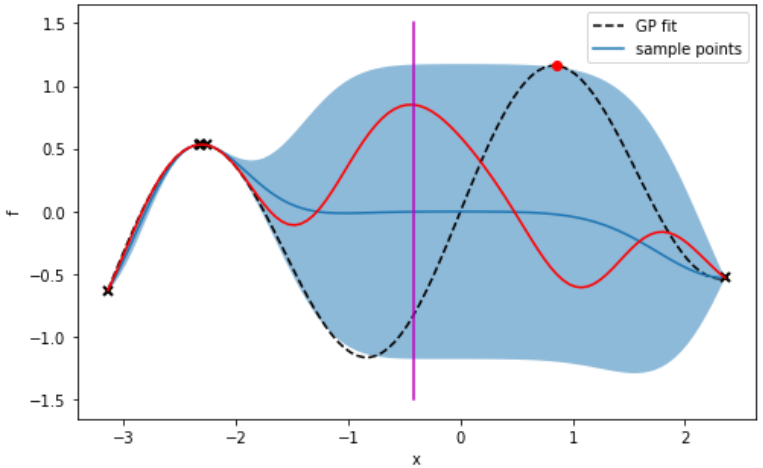
- $\alpha(x) = \tilde{f}(x)$
- $x^* = argmax\big(\alpha(x)\big)$

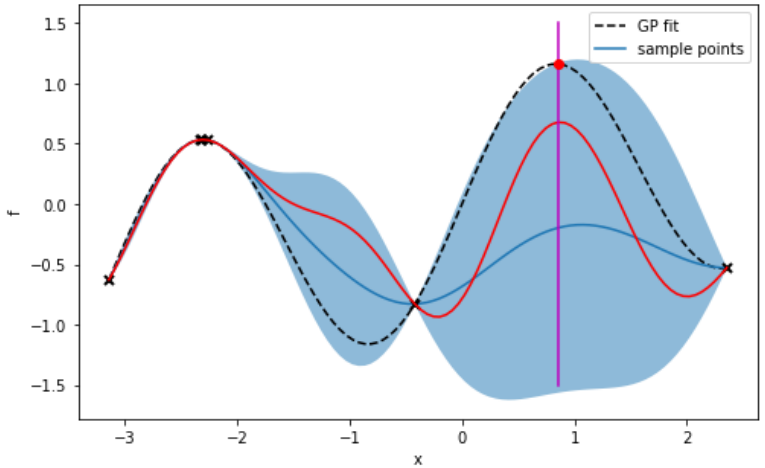
- Mix of Exploitation & Exploration

- GP -> $\mathrm{N}(\mu(x), \sigma(x)^2)$



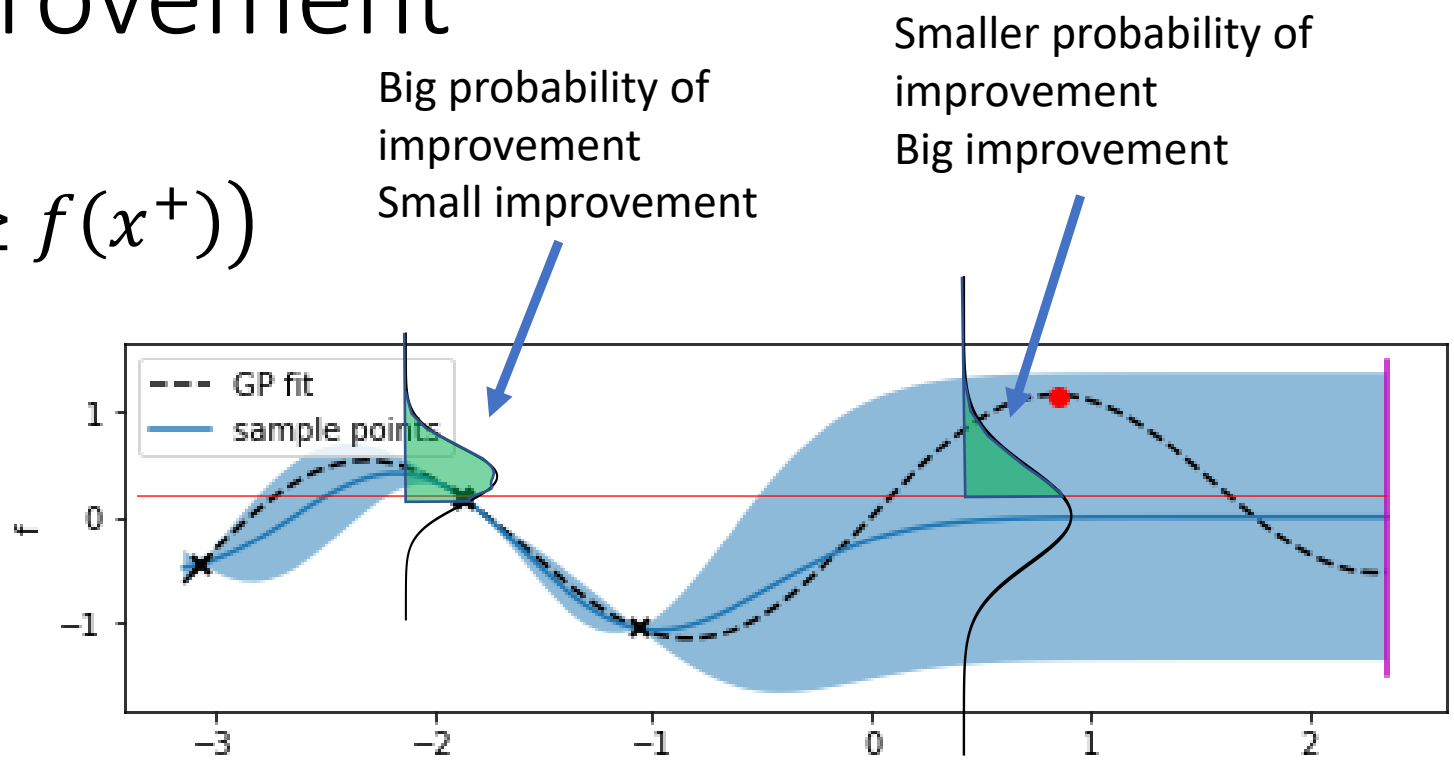






Probability of Improvement

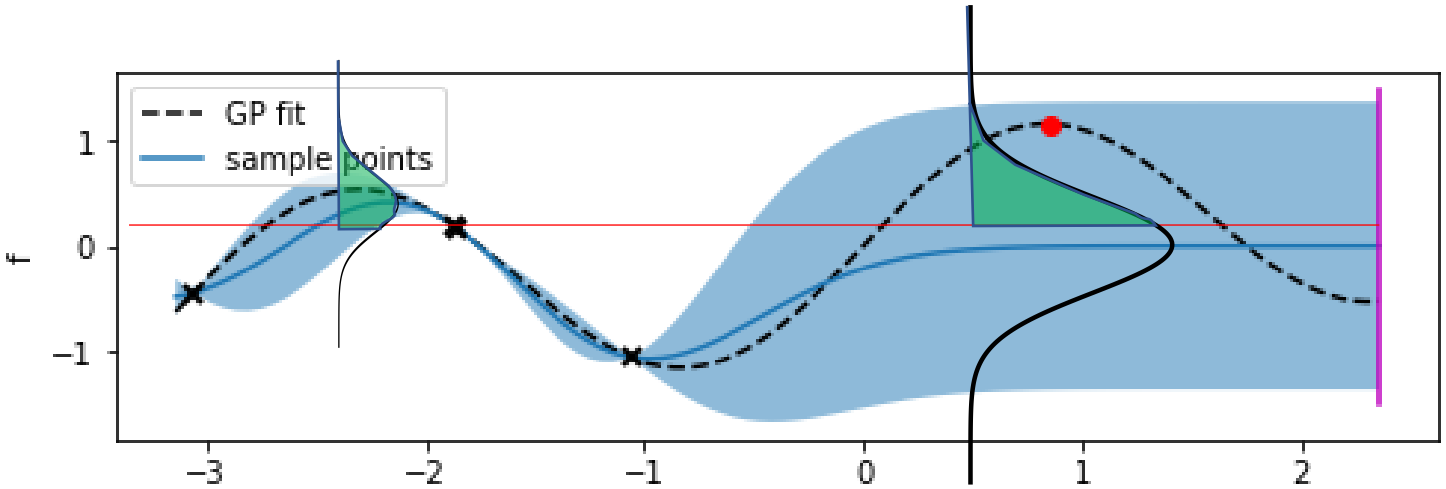
- $\alpha(x) = PI(x) = P\big(f(x) \geq f(x^+)\big)$
- $x^* = argmax\big(PI(x)\big)$



- Maximizes probability of improvement.
- Doesn't care how big the improvement is.
- Can have an x with high probability of improvement over x^+, but not a likely significant improvement.

Expected Improvement

- $\alpha(x) = E(\max\{0, f_{t+1}(x) - f(x^+)\}|\{X, Y\})$
- $x^* = argmax\big(\alpha(x)\big)$



- There is an analytical solution.

Performance Metric: Minimum Regret

- Goal: Find the maximum
- Assume we know the correct answer,

  how do we measure success?
- Min regret: Global max – max (given set)

Continuous example, Global max = 2.2

Iteration 1: “Give me y for x = 50” [50, -0.5]

Min regret = 2.2 – -0.5 = 2.7

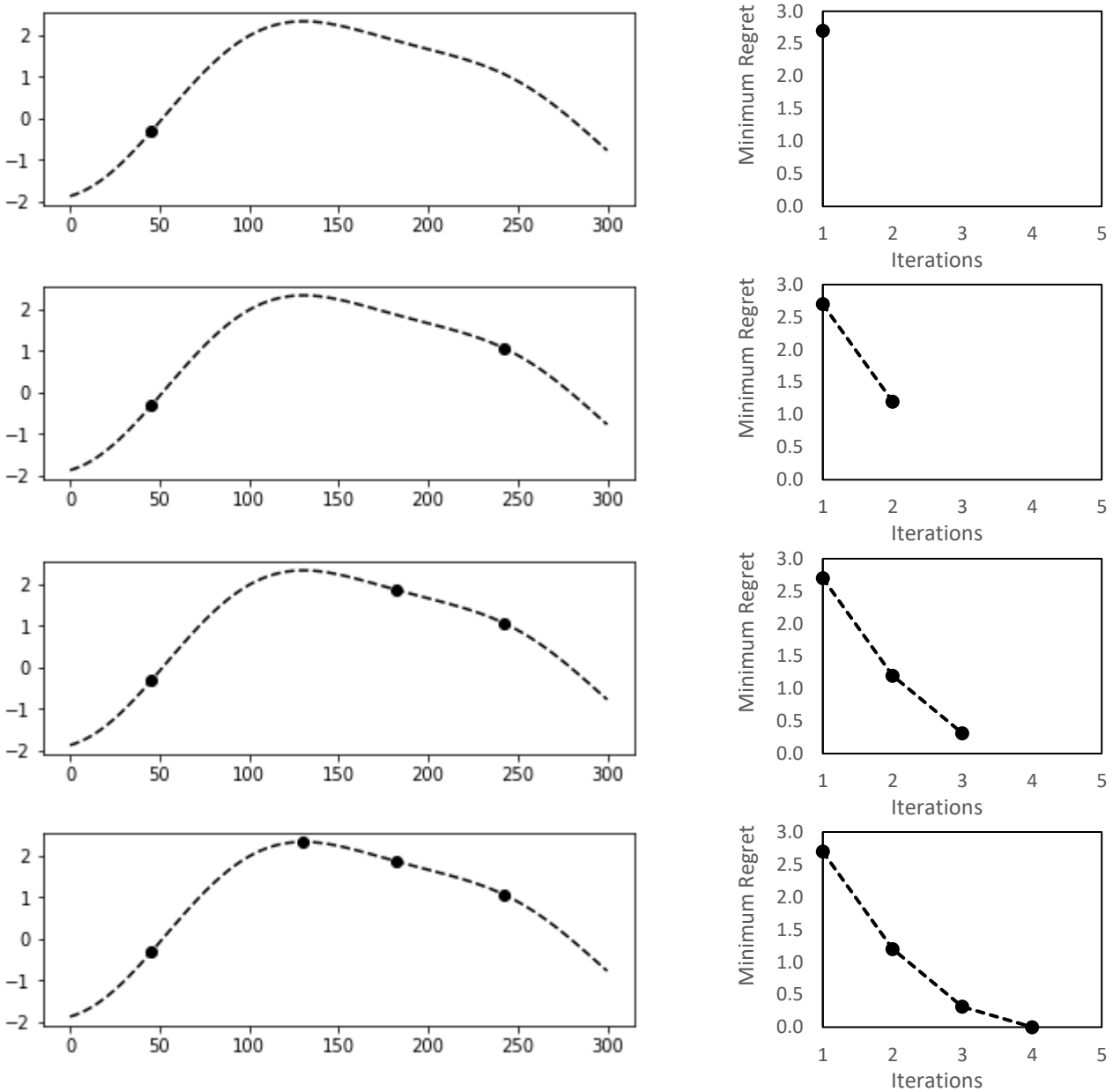
Iteration 2: “Give me y for x = 250” [250, 1]

Min regret = 2.2 – max{-0.5, 1} = 1.2

…

Iteration 4: “Give me y for x = 130” [130, 2.2]

Min regret = 2.2 – max{-0.5, 1, 1.9, 2.2} = 0

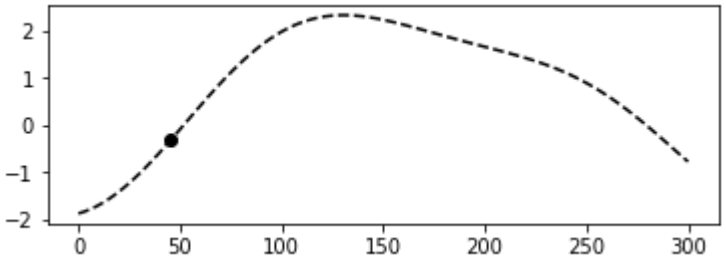


Performance Metric: Convergence

- What if we don't know the correct answer?

$y = \{0.5, 1, 1.9, 2.2, 2.2, 2.2, \ \ldots\}$

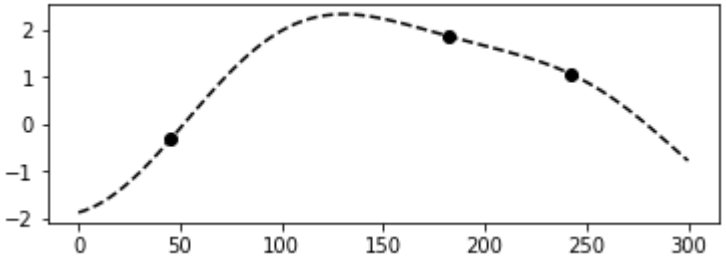
$\mathrm{x} = \{50, 250, 180, 130, 130, 130, \ \ldots\}$



- What if we don't know the correct answer?

$y = \{0.5, 1, 1.9, 2.2, 2.2, 2.2, \ \ldots\}$

$\mathrm{x} = \{50, 250, 180, 130, 130, 130, \ \ldots\}$



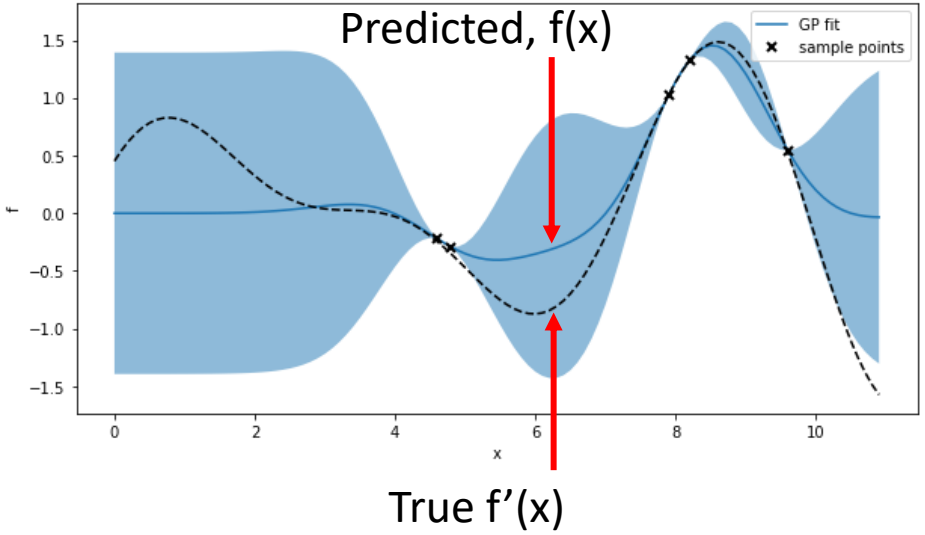
- What if we don't know the correct answer?

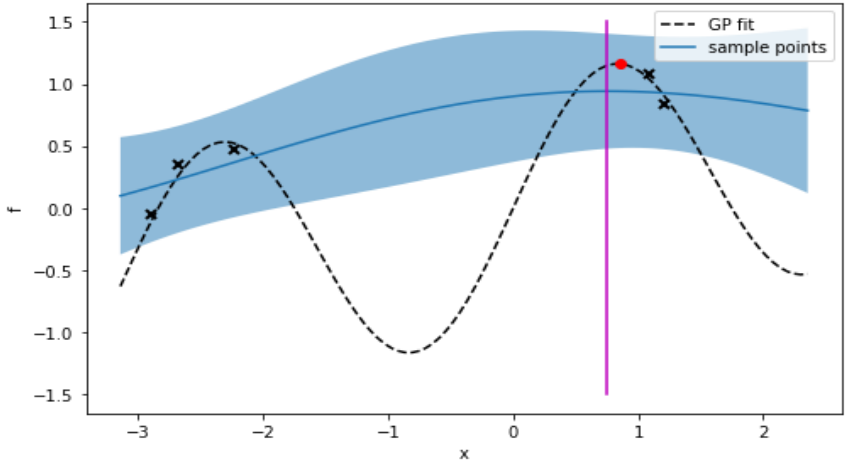
$y = \{0.5, 1, 1.9, 2.2, 2.2, 2.2, \ \ldots\}$

$\mathrm{x} = \{50, 250, 180, 130, 130, 130, \ \ldots\}$

Performance Metric: Analysis of Residuals

- Goal: To fully map the function
- Assume we know the correct $f'(x)$, how do we measure success?
- $Residual = |f(x) - f'(x)|$
- $RMSE = \sqrt{\sum_x |f(x) - f'(x)|^2}$
- Otherwise, can use convergence!



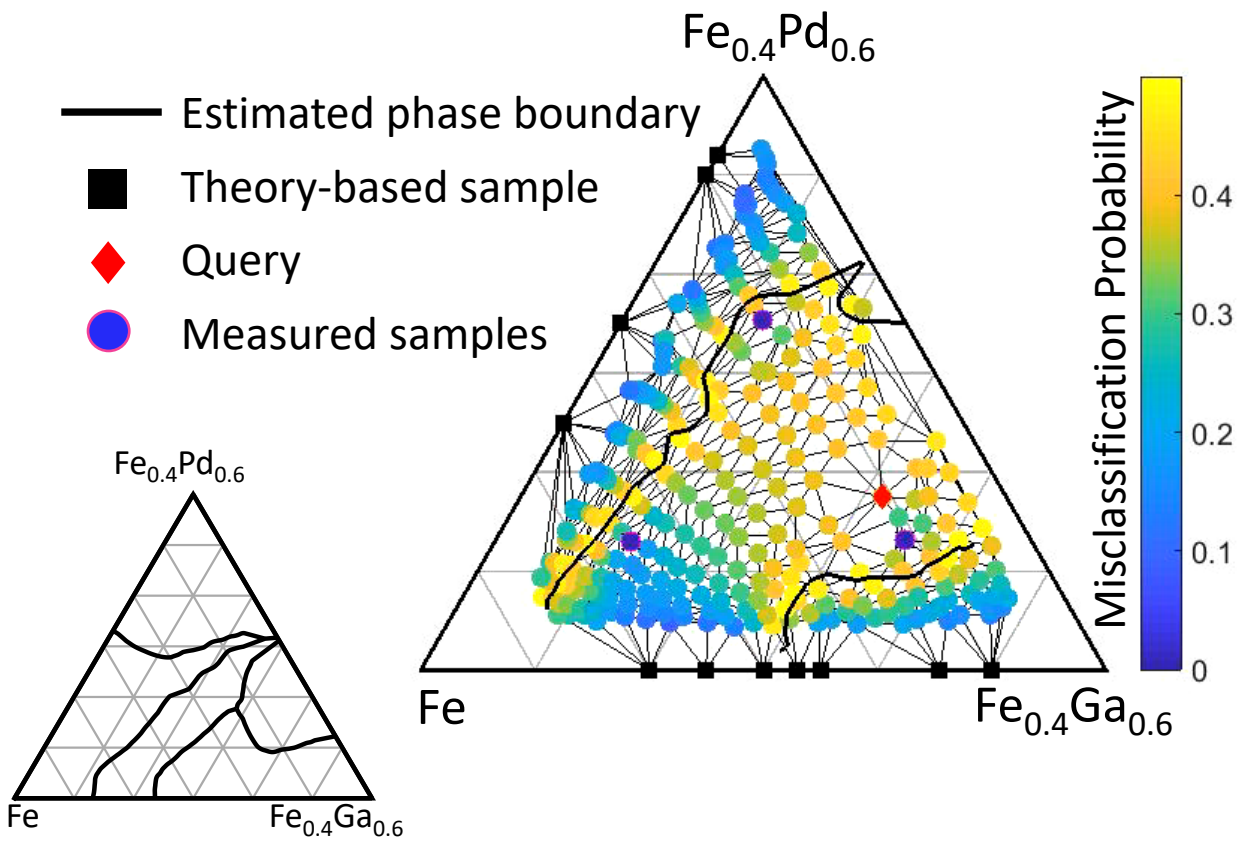


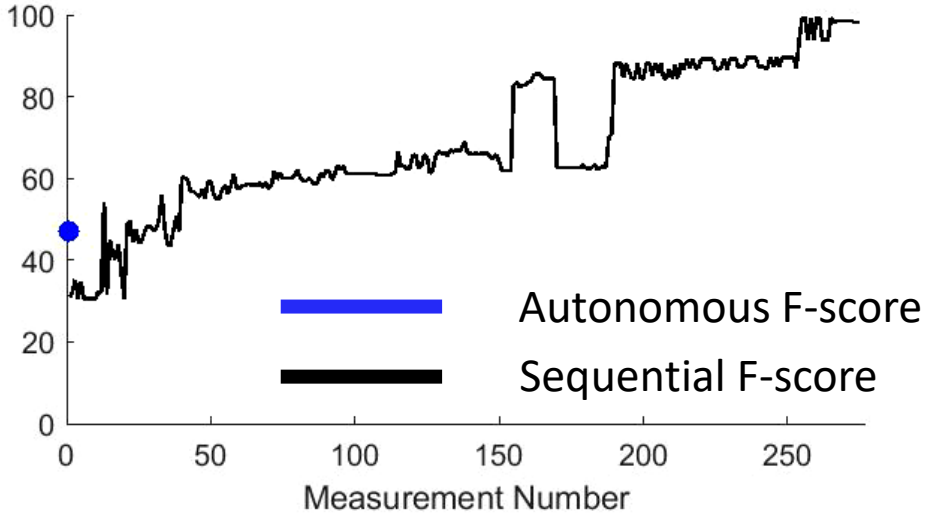
Performance Metrics: Multiple Runs

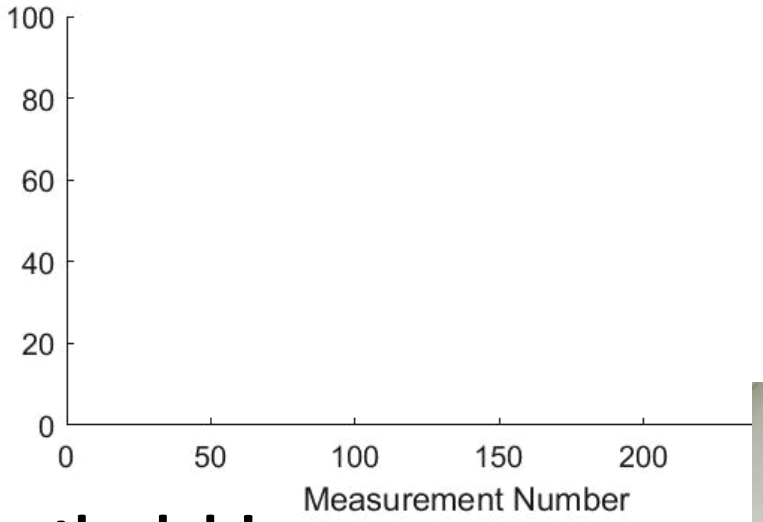
- Mean and Variance of performance

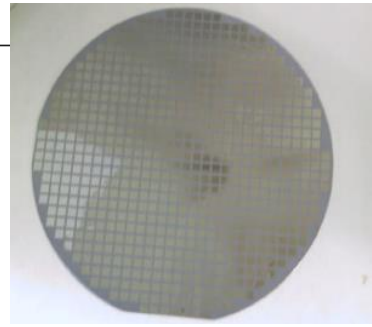
- See Jupyter notebook

Autonomous Phase Mapping





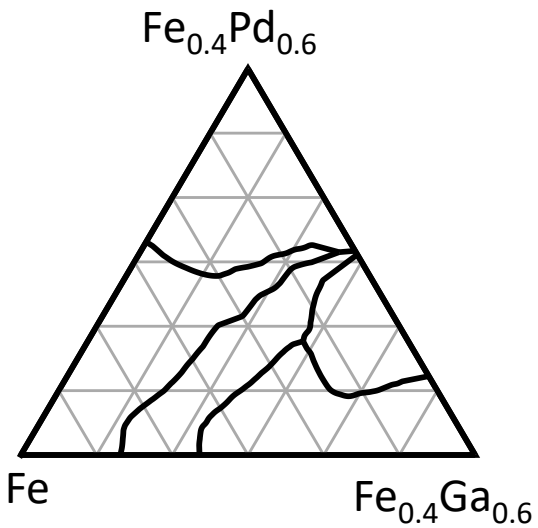




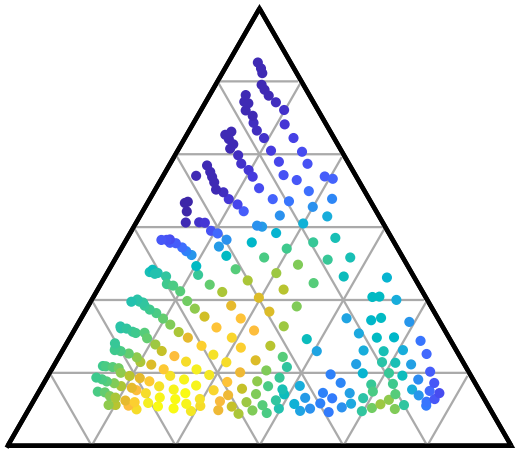
AI is controlling X-ray diffraction systems at SLAC & in the lab!

Phase Diagram + Functional Property Optimization

- Crystal structure impacts functional properties.
- Exploit phase diagram to hone in on optimal materials.
  - At each iteration, measure XRD and Functional Property

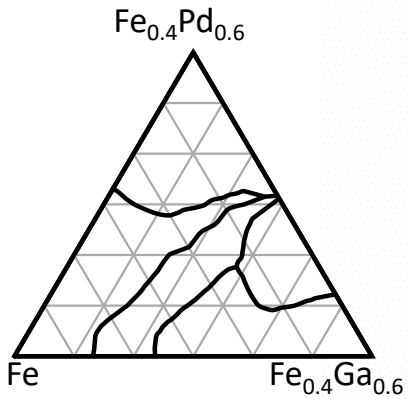


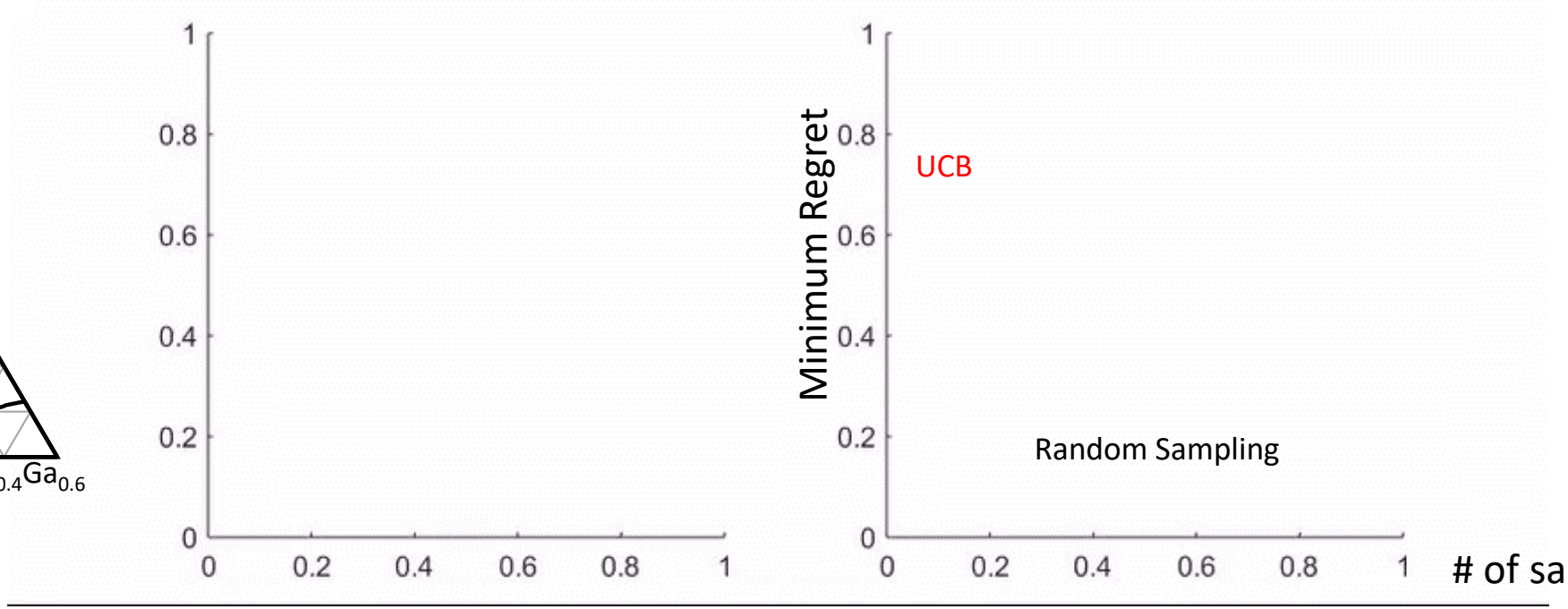
Remnant Magnetization



Functional Property Optimization - Phase Map Informed

- At every measurement Collect XRD and Magnetization

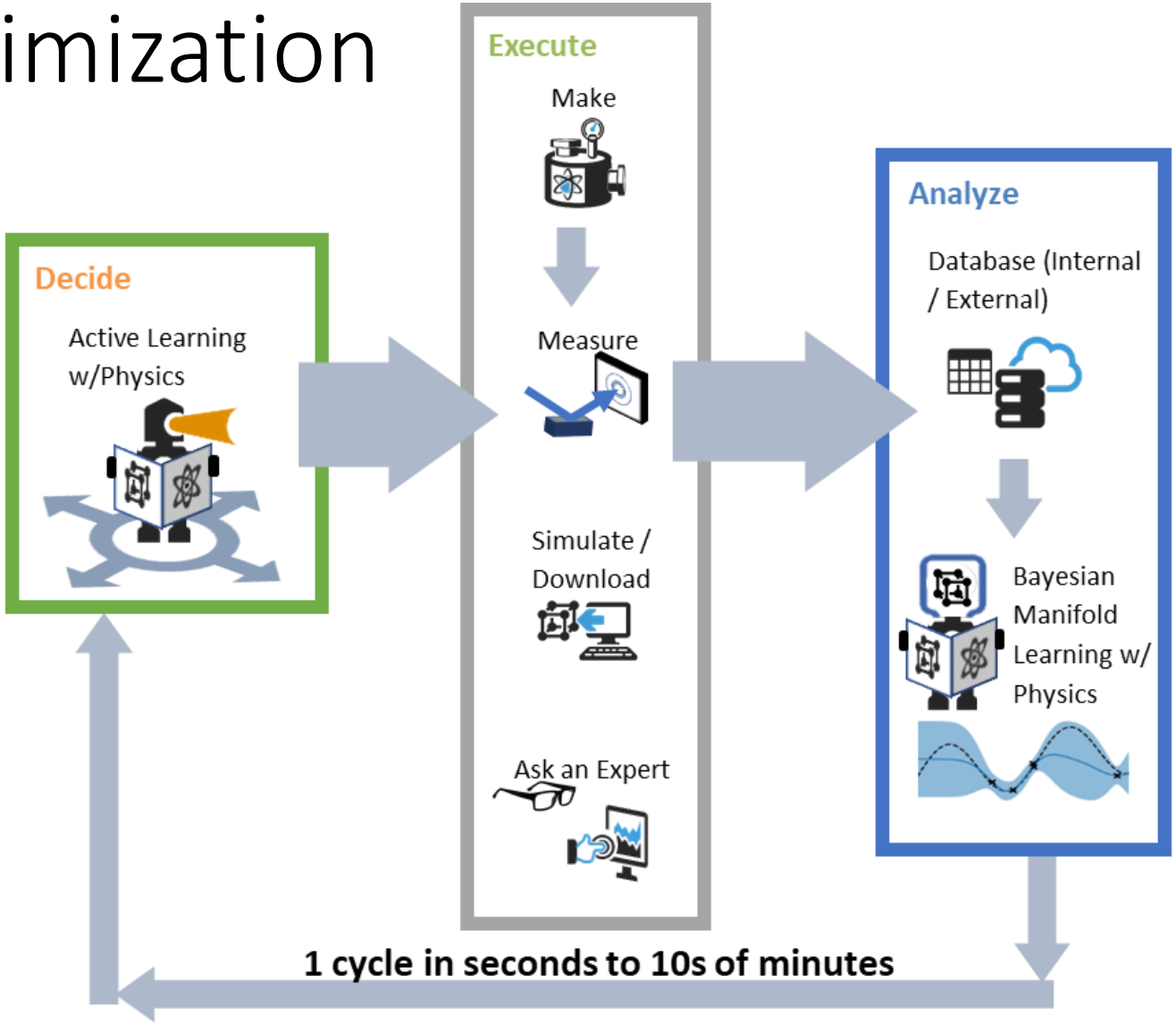




# of samples measured

CAMEO: Closed-Loop Autonomous Materials Exploration and Optimization

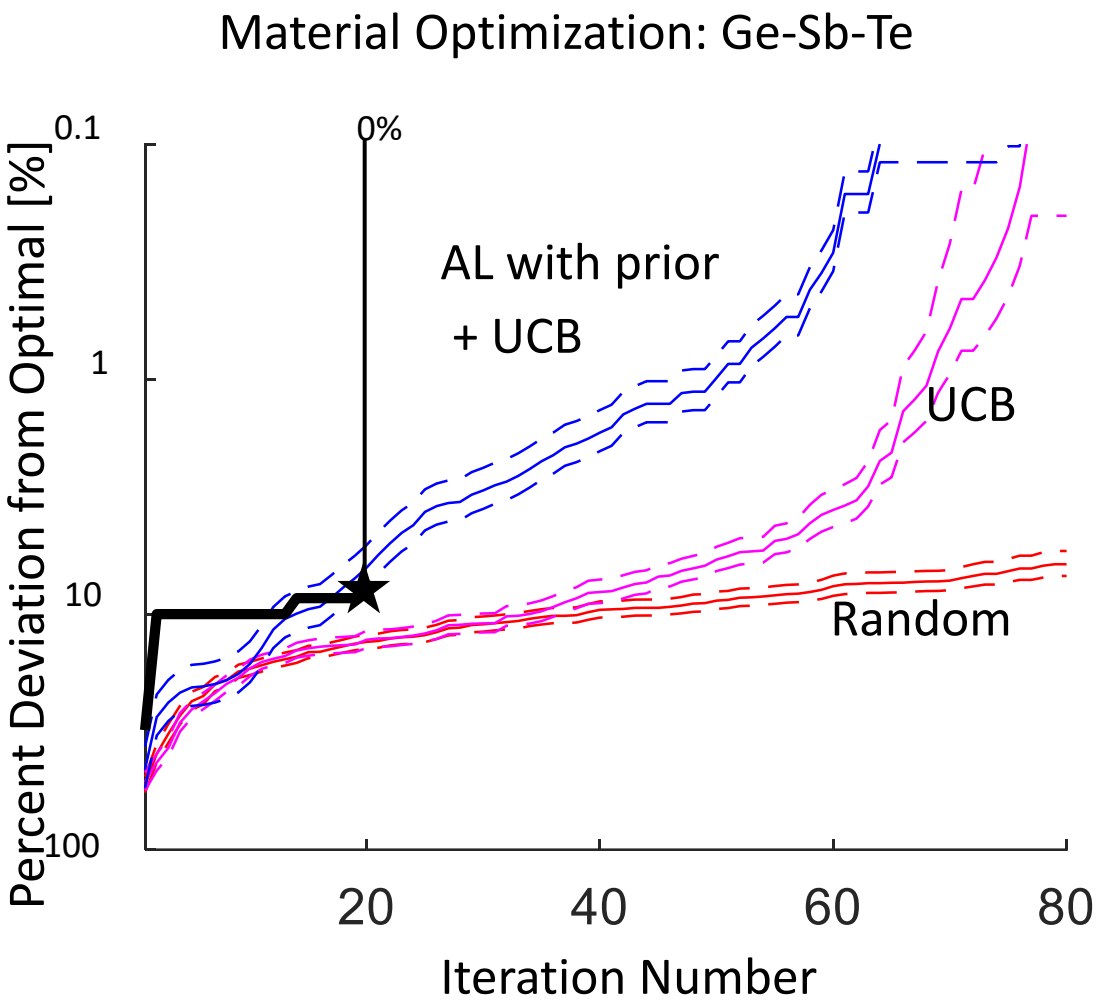
- Improved Speed & Accuracy:
  - Optimal experiment design
  - Encoded prior knowledge
  - Access to external/internal DBs
  - Instrument control
- Interpretability + Uncertainty
  - Bayesian method
  - Visualizations



Kusne, et al. "On-the-fly closed-loop materials discovery via Bayesian active learning." Nature Communications 11.1 (2020)

CAMEO: Find best phase change material

- 10x faster Exploration & Discovery
- New material discovered.
  - Novel nanocomposite phase change memory material
  - Superior to previous best material.



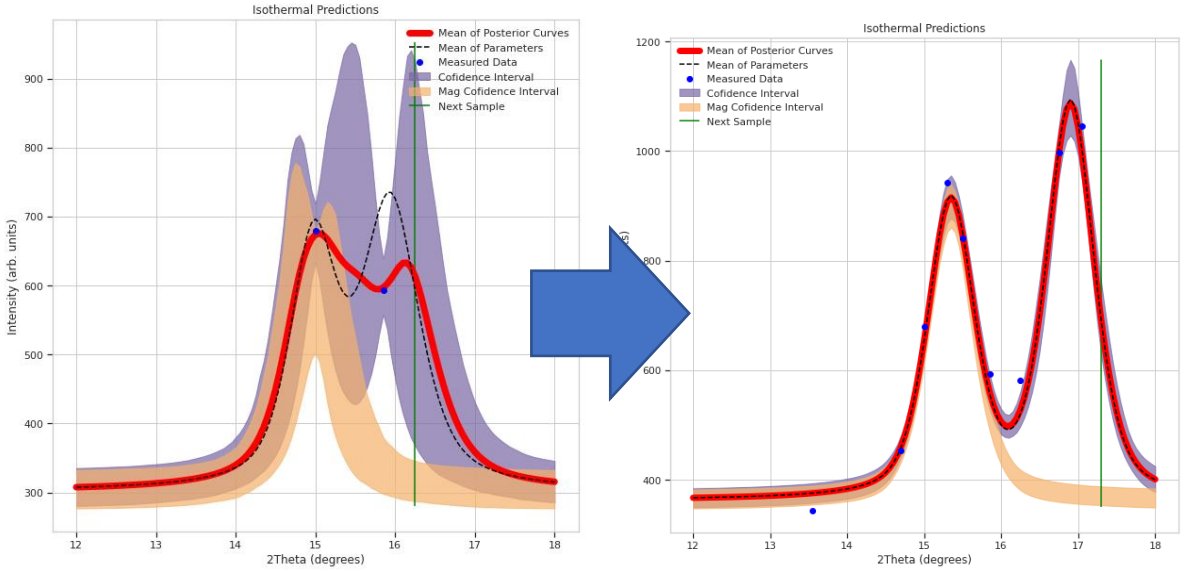
Kusne, et al. "On-the-fly closed-loop materials discovery via Bayesian active learning." Nature Communications 11.1 (2020)

Autonomous Neutron Scattering: Superconductors

- Autonomous control: 2Theta and Temperature

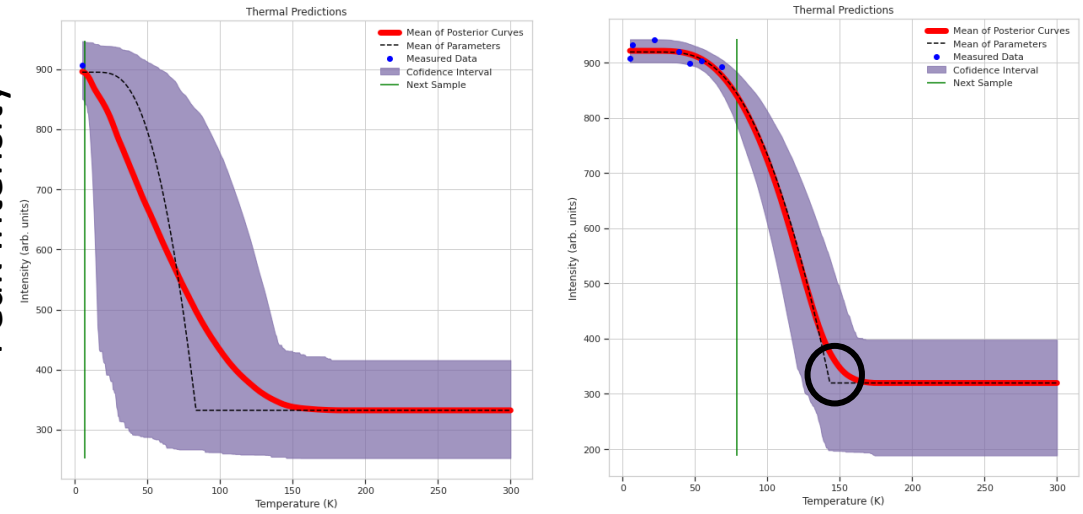
Austin McDannald

Peak Parameters



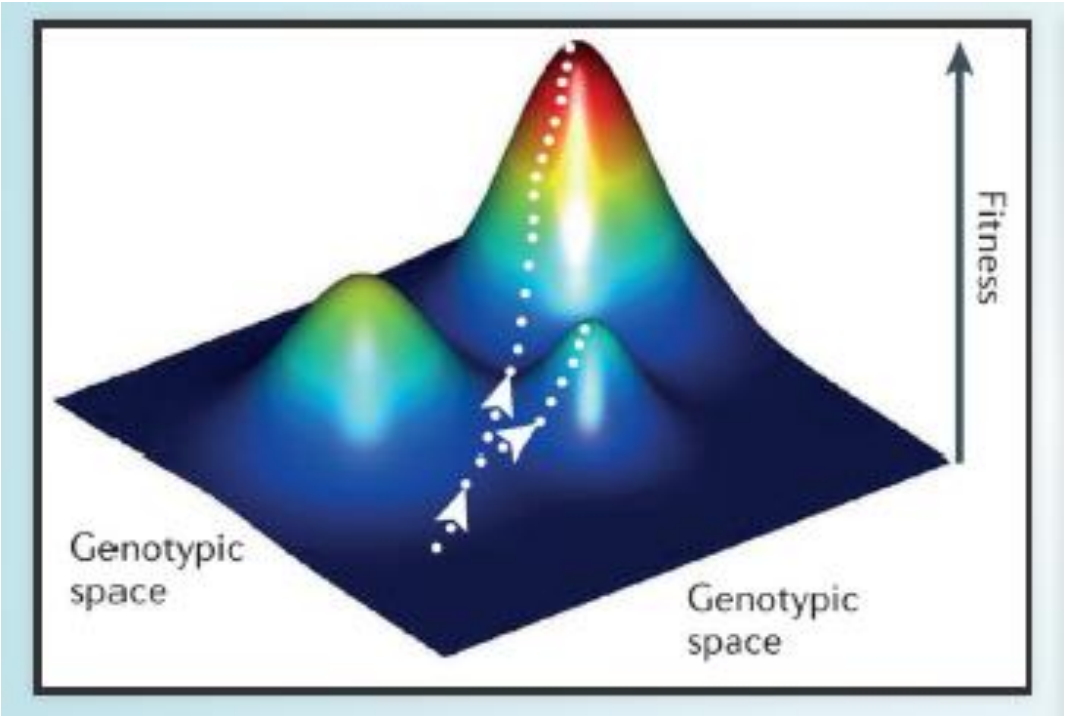
Neel Temperature

Peak Intensity



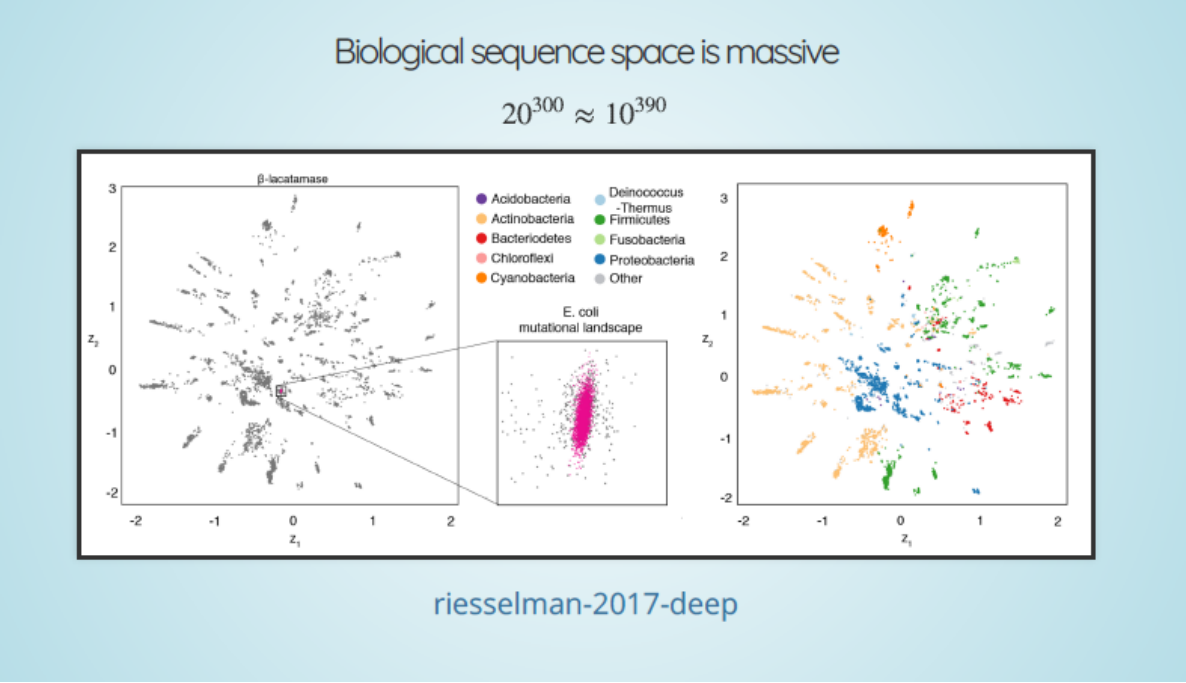
Temperature

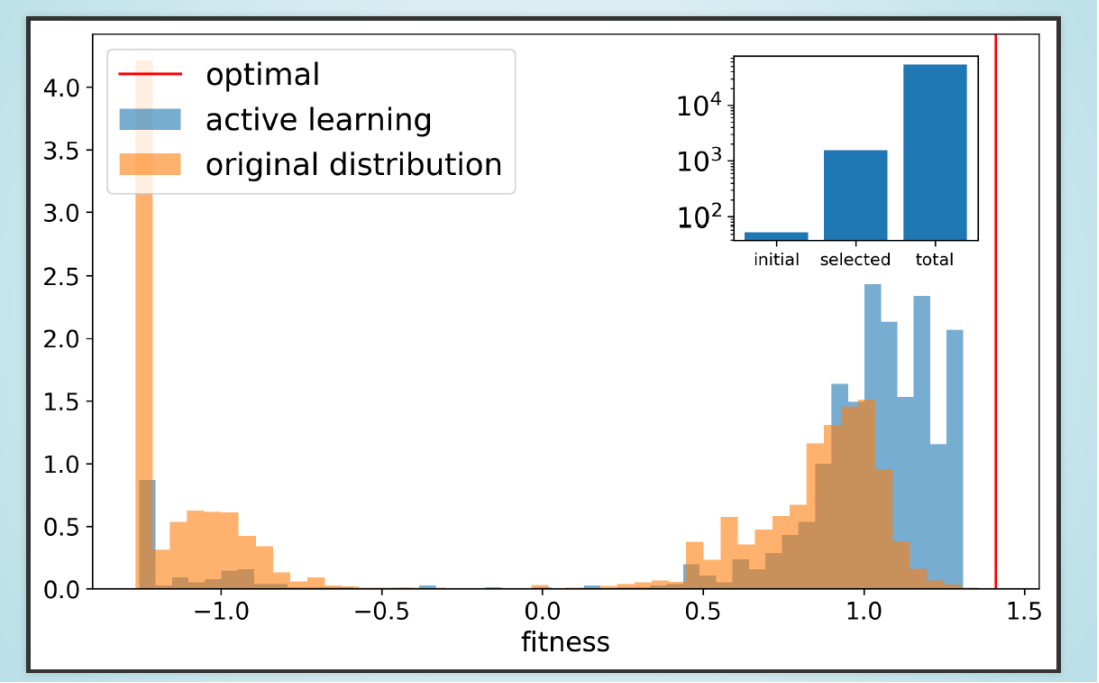
Autonomous System for Synthetic Biology



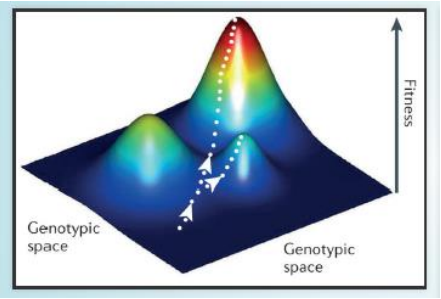
Peter Tonner, et al. AAAI Synthetic Biology 2018

Autonomous System for Synthetic Biology





Peter Tonner



Peter Tonner, et al. AAAI Synthetic Biology 2018

Autonomous System for Synthetic Biology

NRC Postdoc Openings!

- Search for “NIST NRC Postdoc”

RO #	Program	Title
50.64.31.B8558	NIST	Machine Learning for Autonomous Genetic Engineering of Microbial Systems
50.64.31.B8265	NIST	Machine Learning for High Throughput Materials Discovery and Optimization Applications
50.64.31.B8559	NIST	Machine Learning-driven Autonomous Systems for Materials Discovery and Optimization
•50.68.51.C0577	NIST	Machine Learning Driven Autonomous Metrology System

Infinitely large NN -> GP (high level)

- Linear Model: $z = \beta_0 + \sum_j x_j \beta_j$
  - Going to modify this slightly so that it is the output of a single layer NN

- $z^1(x) = \boldsymbol{\beta_0^1} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} \boldsymbol{x_j^1}(\boldsymbol{x})\boldsymbol{\beta_j^1}$
  - z^1 is our output
  - x is our observed input data and $x_j^1(x) = \phi\left(\beta_j^0 + \sum_k x_k \beta_{jk}^0\right)$ is our activation
  - Assume β_0^1 and β_j^1 are independent & randomly drawn w/ mean 0 and var σ_b^2 and σ_β^2/N
    - Let $\frac{\widetilde{\boldsymbol{\beta}}_j^1}{N} \sim \boldsymbol{\beta}_j^1$ with $\widetilde{\boldsymbol{\beta}}_j^1$ having mean 0 and variance σ_β^2

- Then $\lim_{N\to\infty} \sum_{j=1}^{N} \boldsymbol{x}_j^1(\boldsymbol{x})\boldsymbol{\beta}_j^1 = \lim_{N\to\infty} 1/N \sum_{j=1}^{N} \boldsymbol{x}_j^1(\boldsymbol{x})\widetilde{\boldsymbol{\beta}}_j^1 \to \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{K})$ Central Limit Theorem

- Put another way: at initialization of the NN
  - the output of each layer is comprised of a bunch of independently distributed RV's w/ mean 0 and std. dev σ_β^2
  - CLT states that if we average and infinite number of these it will converge to a Gaussian