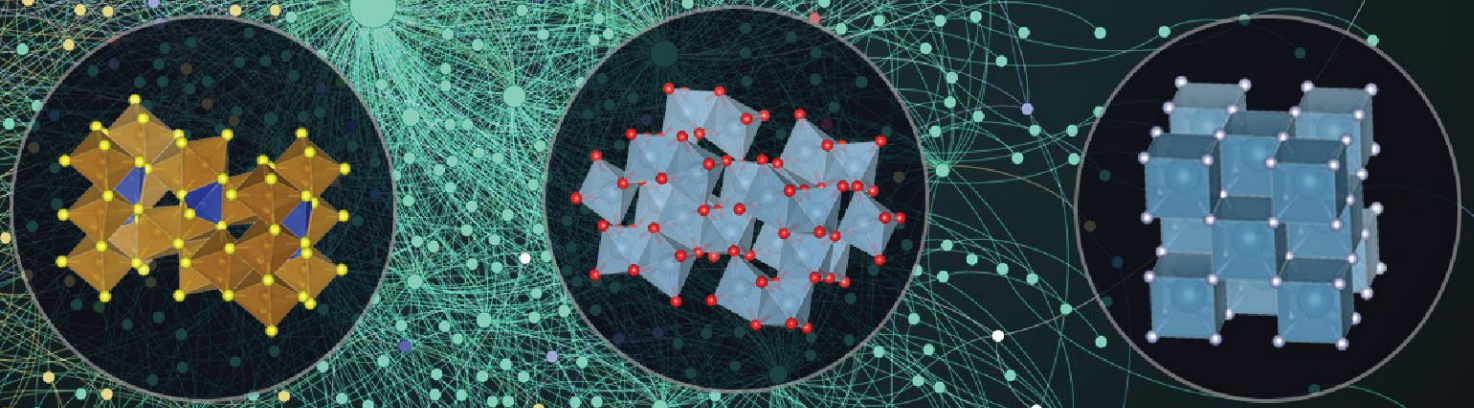
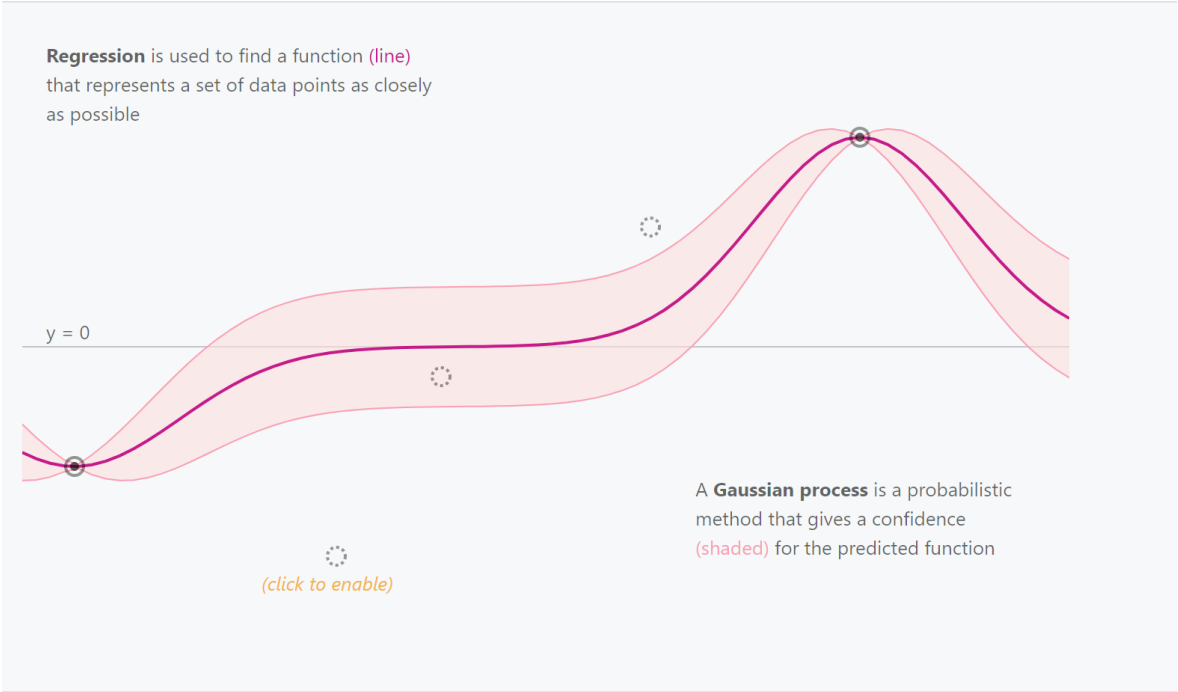
Gaussian processes



Gaussian Processes are a bit tricky to understand!

A Visual Exploration of Gaussian Processes

How to turn a collection of small building blocks into a versatile tool for solving regression problems.



AUTHORS	AFFILIATIONS	PUBLISHED	DOI
Jochen Görtler	University of Konstanz	April 2, 2019	10.23915/distill.00017
Rebecca Kehlbeck	University of Konstanz		
Oliver Deussen	University of Konstanz		

https://distill.pub/2019/visual-exploration-gaussian-processes/

IntuitiveML (YouTube) has a pretty good walkthrough and summary

- Input: training data (TRAIN) and test data (TEST)
- Output: value and confidence interval at every point in TEST

How it works:

- Center the data to have mean 0
- Concatenate the TRAIN and TEST datapoints to COMBINED
- Create a covariance matrix
  - Use a kernel function to find similarity between each point in COMBINED
  - The similarity between point *i* and point *j* is the value of the matrix at (*i,j*)
- Calculate the joint distribution using the covariance matrix
- Calculate the conditional distribution – aka calculate the probability of TEST given TRAIN
- Marginalize the conditional distribution – aka calculate the probability of distribution of a single point in TEST given all other points
- Obtain mean and stdev for every point in TEST
- Calculate the confidence interval using stdev to the desired level (eg 95%)
- Mean is the regression value

IntuitiveML (YouTube) I get confused trying to learn Gaussian Processes | Learn with me!

In general, we are trying to predict Y conditional on X

$$C(Y|X) = g(X)$$

Consider linear regression

$$C(Y|X) = g(X, \beta) = \beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots$$

Assumptions:

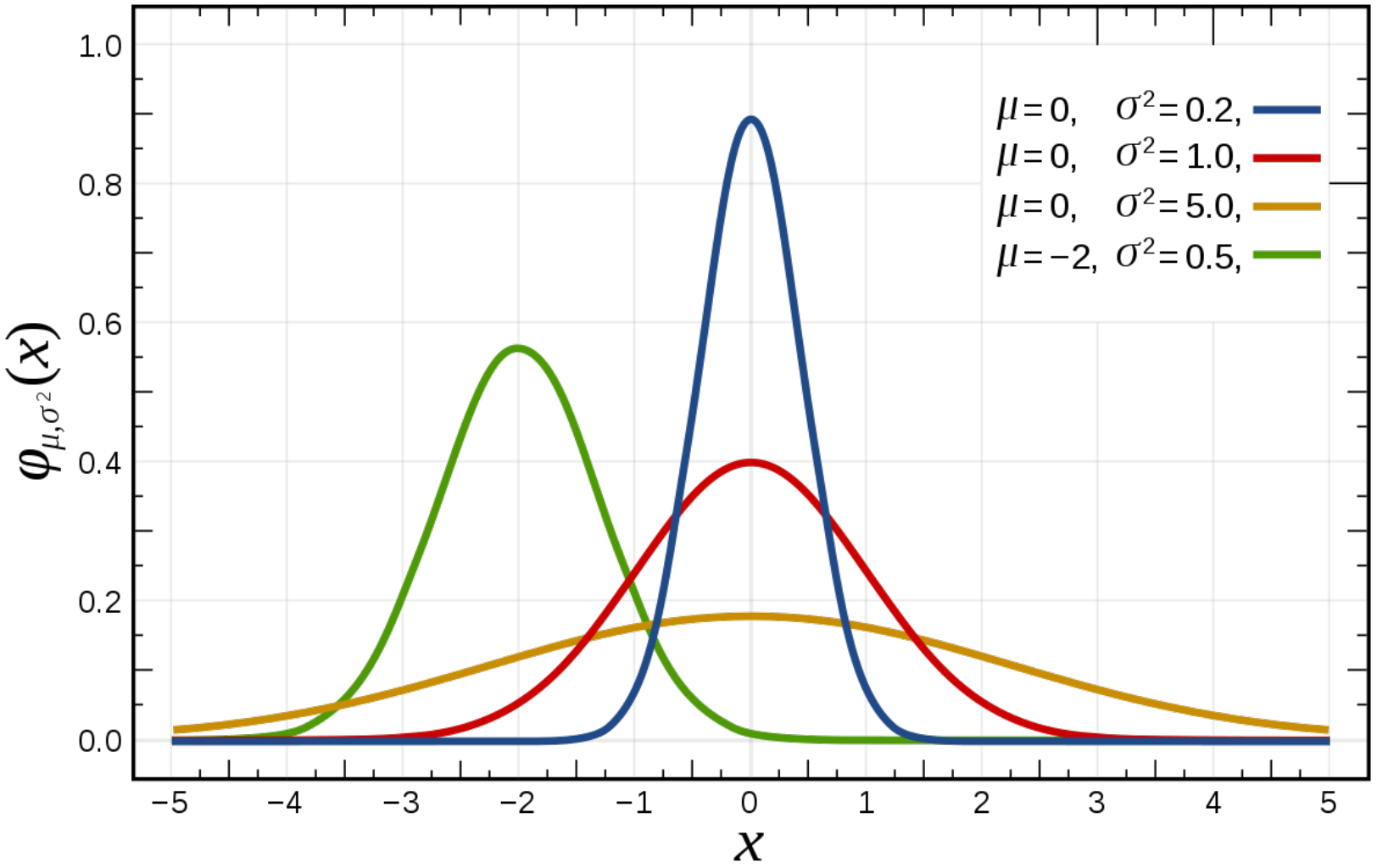
Parametric (data is normally distributed) and linear relationships between variables

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma)$$

Logistic regression also possible

$$C(Y|X) = \frac{1}{1 + \exp(\beta X)}$$

Instead of relying on linear or non-linear approaches to fit data, let's rely on gaussians!



https://en.wikipedia.org/wiki/Normal_distribution

Instead of relying on linear or non-linear approaches to fit data, let's rely on gaussians!

Consider a bivariate Gaussian distribution

$$p(y_1, y_2|\mu, \Sigma) = N\left(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\sigma_1^2 & \sigma_1\sigma_2\rho\\ \sigma_1\sigma_2\rho & \sigma_2^2\end{bmatrix}\right)$$

Off diagonal elements in standard deviation matrix are the covariance terms which include the correlation term ($\rho = 0, uncorrelated, \rho = 1, correlated$)

https://en.wikipedia.org/wiki/Normal_distribution

Gaussians can be easily marginalized and conditioned

Why are Gaussians a great idea?

1. They are easy to marginalize (remove all parts of covariance matrix except what you want to keep)
   - if you want to ditch y_2, then drop $\mu_2, \sigma_1\sigma_2\rho, and\ \sigma_2^2$

2. They are easy to condition (given the value of one value in our multivariate normal distribution, what do we expect some other unobserved value to be?)
   - $p(x|y) = N\big(\mu_x + \Sigma_{xx}\sum_y^{-1}\big(y - \mu_y\big)\ \big), \Sigma_x - \Sigma_{xy}\sum_y^{-1}\Sigma_{xy}^T$

     Conditional mean , conditional covariance
   - Where μ_x is marginal mean (easy) and Σ_x is marginal covariance each minus some easy to calculate terms

In Gaussian Process we will generalize multivariate normal to contain infinite elements

Gaussian distribution:

$$y \sim N(\mu, \Sigma)$$

Gaussian process:

$$f \sim GP(m(x), k(x, x'))$$

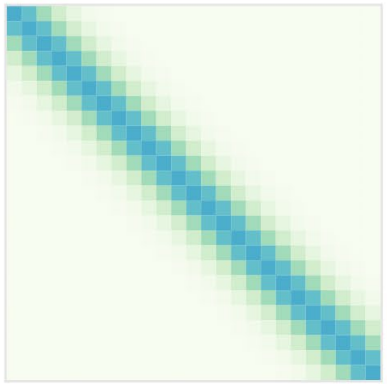
Where $m(x)$ is a mean function, and $k(x, x')$ is a covariance function

Definition: Gaussian process is an infinite collection of variables, any finite subset of which has a Gaussian distribution.

Tricky part of Gaussian process is figuring out which function to use to model covariance

RBF KERNEL

$$\sigma^2 \exp\left(-\frac{||t-t'||^2}{2l^2}\right)$$

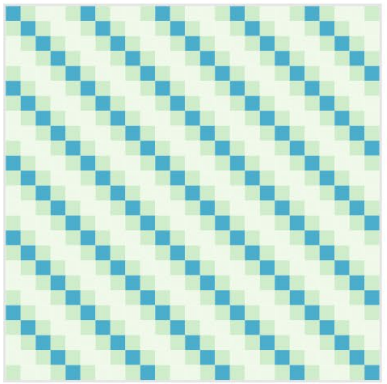


Variance σ = 0.8

Length l = 0.8

PERIODIC

$$\sigma^2 \exp\left(-\frac{2\sin^2(\pi|t-t'|/p)}{l^2}\right)$$



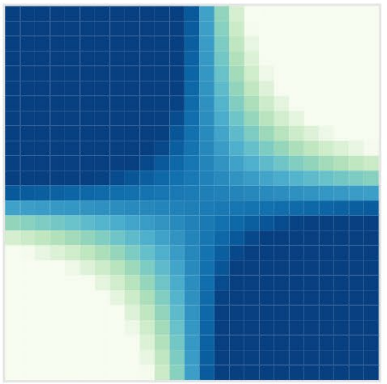
Variance σ = 0.8

Length l = 0.8

Periodicity p = 0.5

LINEAR

$$\sigma_b^2 + \sigma^2(t-c)(t'-c)$$



Variance σ = 0.35

Variance σ_b = 0.8

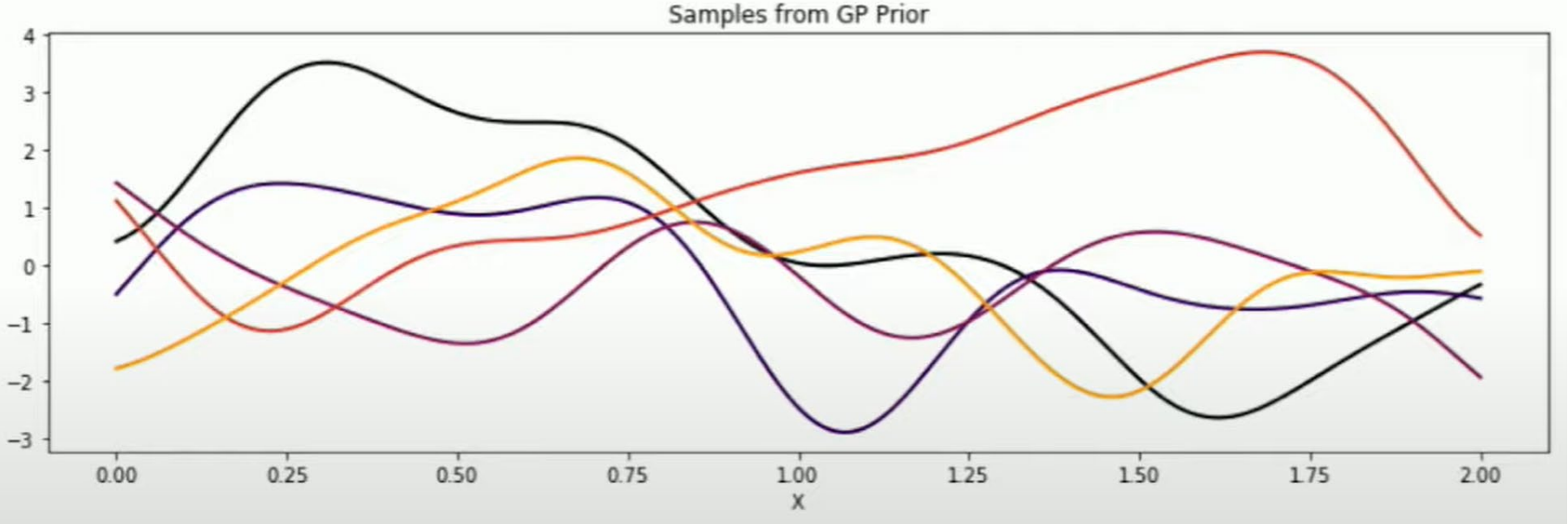
Offset c = 0

For the **Linear** kernel
the parameter **Variance σ** determines
the average distance away from the function's mean.

https://distill.pub/2019/visual-exploration-gaussian-processes/

Consider all the different covariance functions we can get from just one exponential quadratic

$$k(x, x') = \exp\left[-\frac{(x - x')^2}{2\ell^2}\right]$$

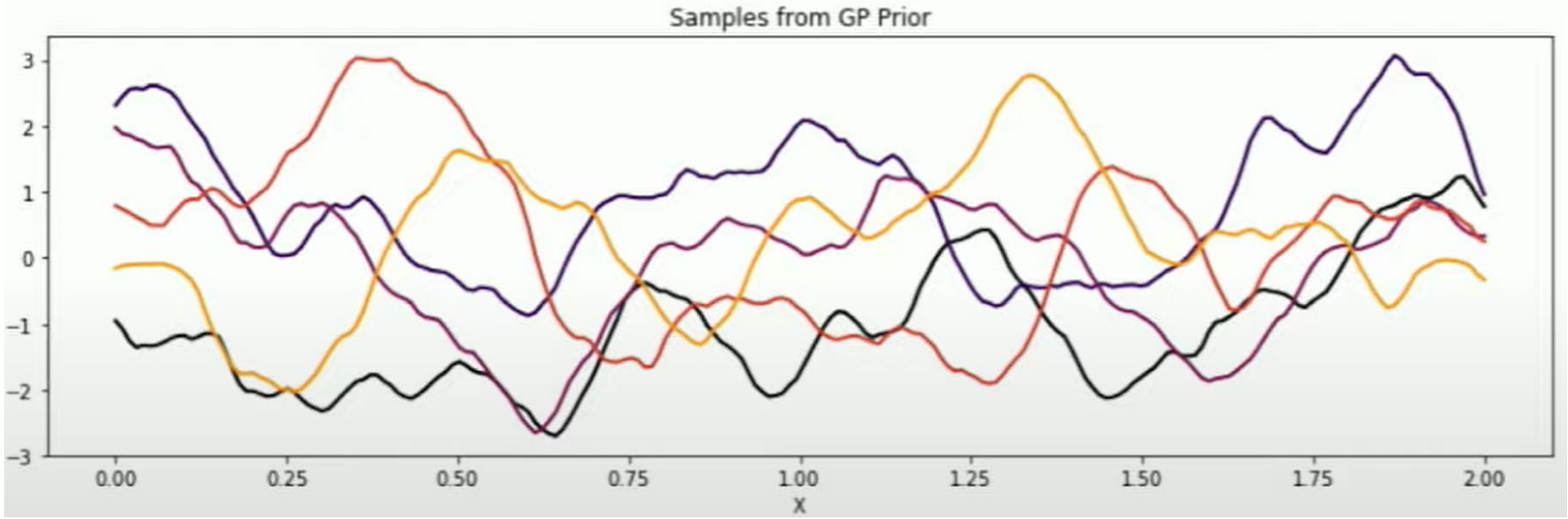


One parameter, length scale, that determines how far away you have to go away from two points to make them independent

Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

Matern(3/2) is another popular covariance function

$$k(x, x') = \left(1 + \frac{\sqrt{3(x - x')^2}}{\ell}\right) \exp\left[-\frac{\sqrt{3(x - x')^2}}{\ell}\right]$$

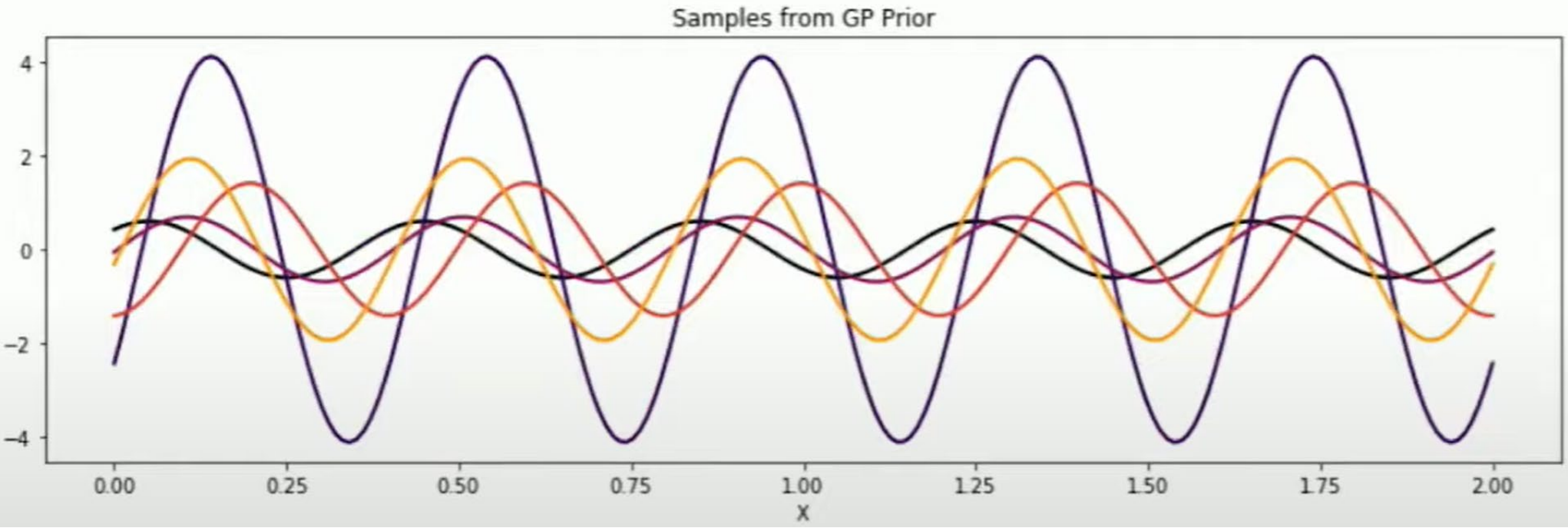


Length scale parameter, plus a scaling term to scale roughness

Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

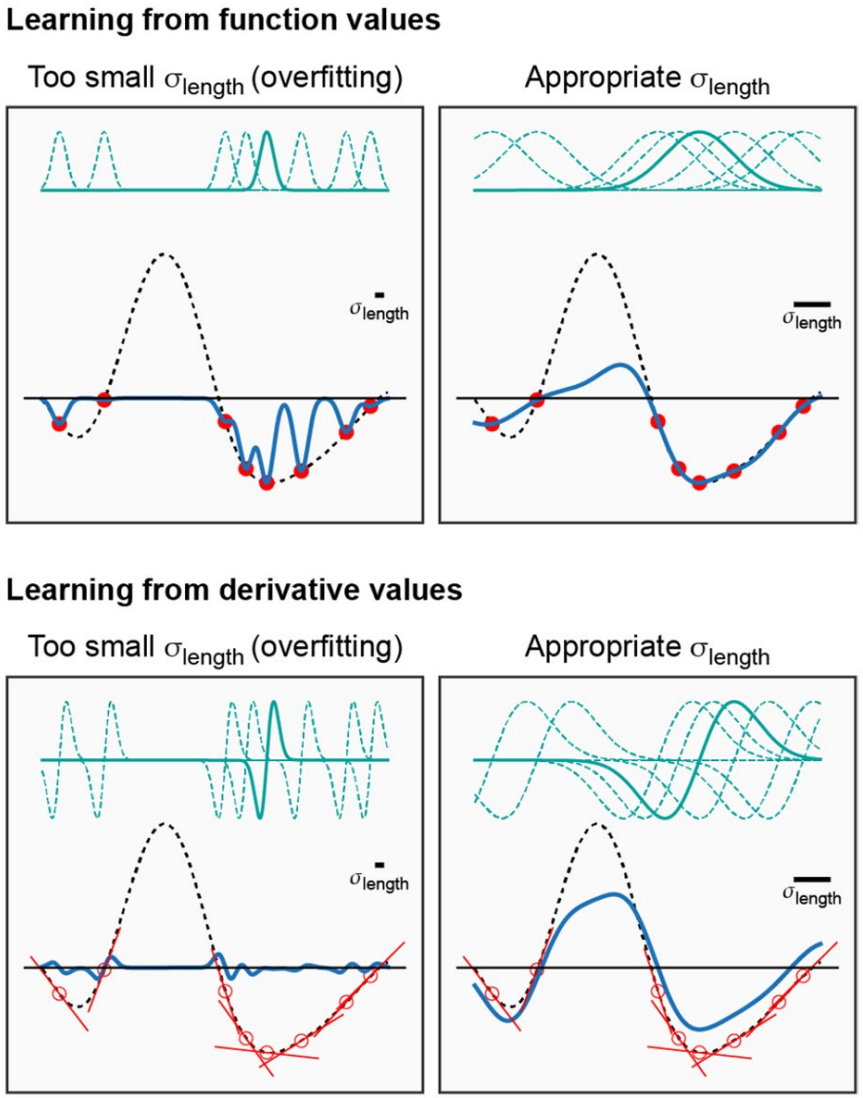
Cosine is another popular covariance function

$$k(x, x') = \cos\left(\frac{||x - x'||}{\ell^2}\right)$$



Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

Getting the kernel hyperparameters right is important



Deringer et al Gaussian Process Regression for Materials and Molecules, Chemical Reviews 2021

A mean function is also necessary, but not as hard to pick

Mean function generates mean vectors

$$\mu \sim m(x|\phi)$$

Simplest is just zero!

$$m(x) = 0$$

Or constant

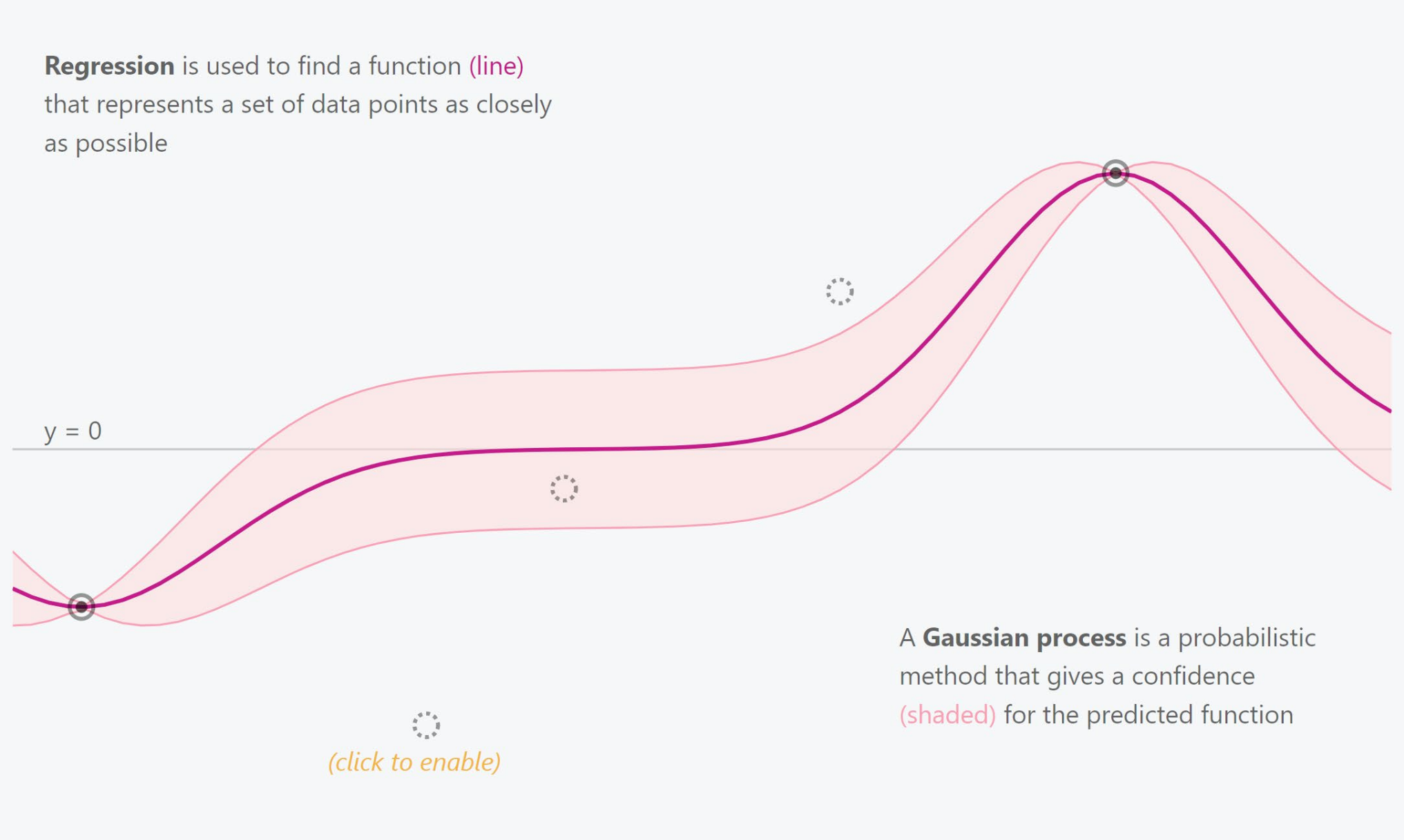
$$m(x) = C$$

Or linear

$$m(x) = Ax + b$$

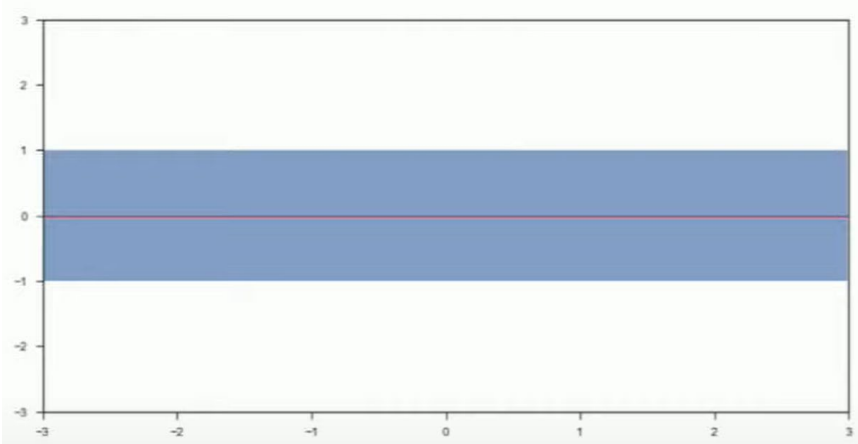
Posterior does not use the mean, it is just used as a guess for the prior because where you have no data, it trends towards the mean

In the absence of data, the regression trends towards the mean

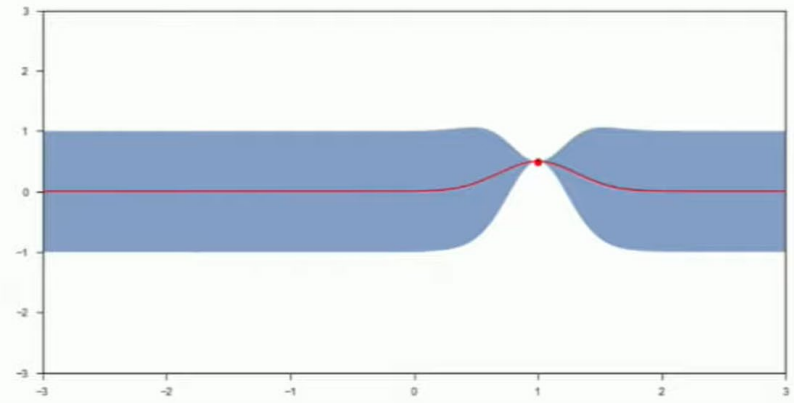


https://distill.pub/2019/visual-exploration-gaussian-processes/

Let’s draw from a prior Gaussian process and we’ll do it point by point

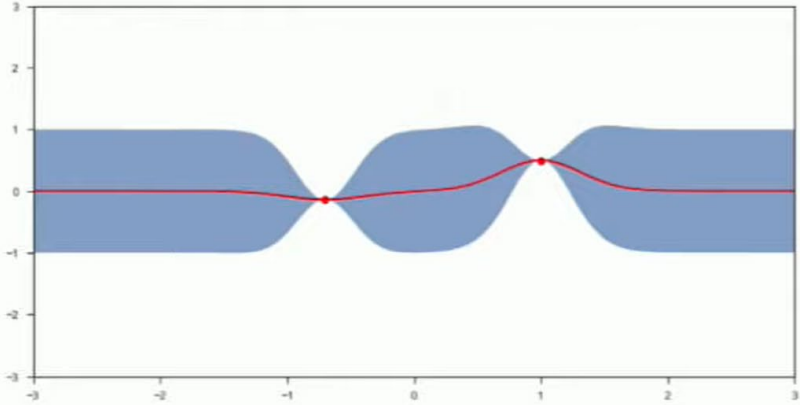


$$y \sim GP(m = 0, k = \text{Quad}(1))$$



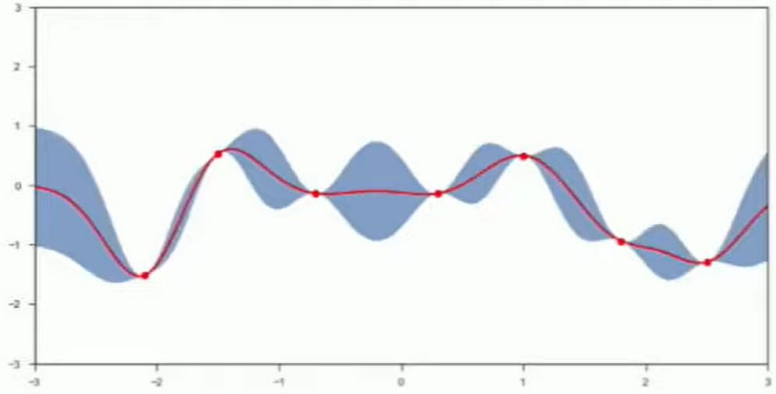
```
x = [1.]
y = [np.random.normal(scale=σ_0)]
y
```

```
[0.4967141530112327]
```



```
x = -0.7
m, s = conditional([-0.7], x, y, θ)
y.append(np.random.normal(m, s))
y
```

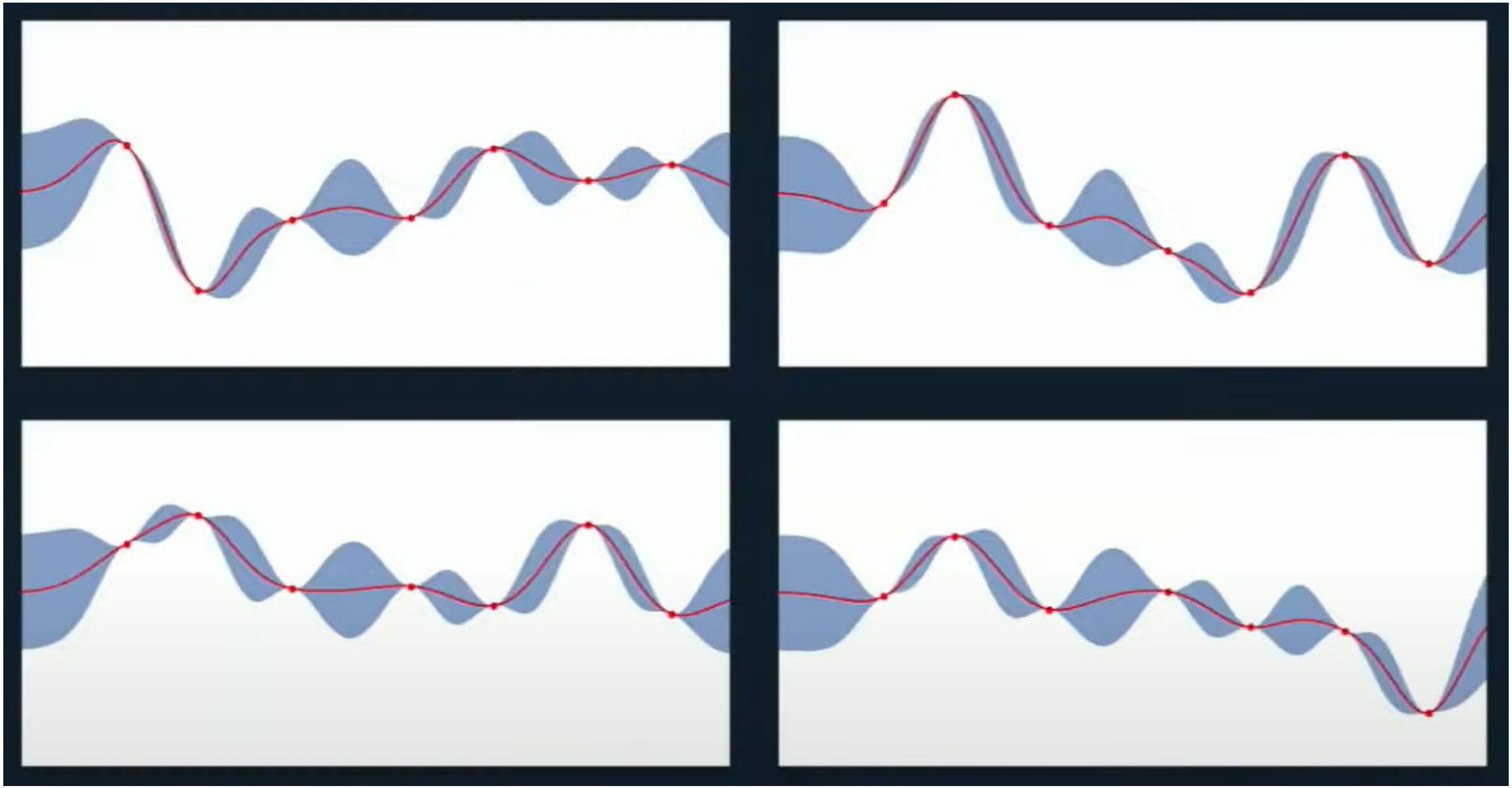
```
[0.4967141530112327, -0.1382640378102619]
```



```
x = [-2.1, -1.5, 0.3, 1.8, 2.5]
mu, s = conditional(x_more, x, y, θ)
y += np.random.multivariate_normal(mu, s).tolist()
y
```

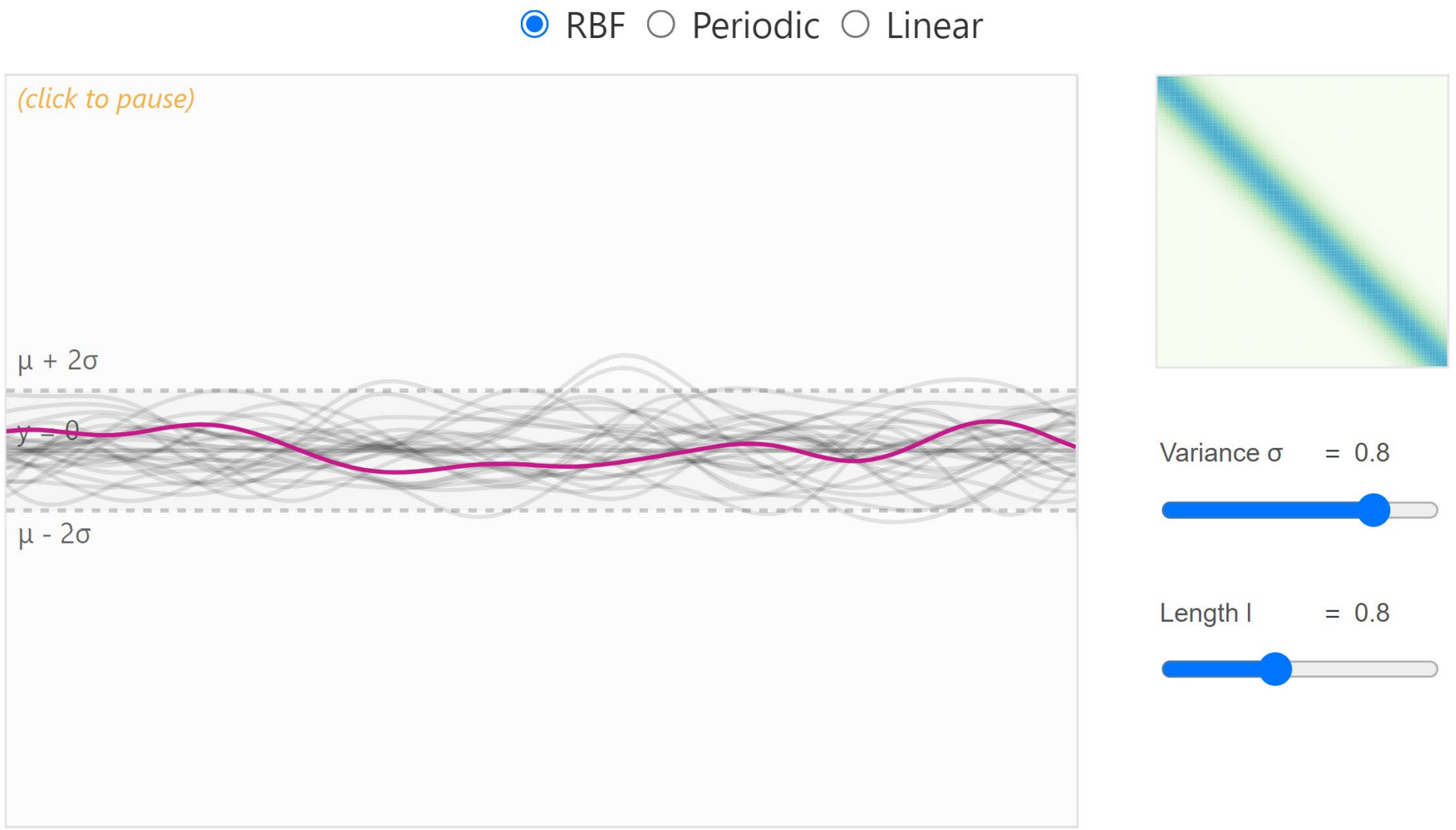
```
[0.4967141530112327, -0.1382640378102619, -1.5128756 ,  0.52371713,
-0.13952425, -0.93665367, -1.29343995
```

We could create as many different functions as we like by sampling different parameters



Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

Nice animation of the sampling of distributions



https://distill.pub/2019/visual-exploration-gaussian-processes/

We now use these priors to calculate posteriors given some data observations

Recall Bayes' formula

$$p(\theta|y) \propto \prod_{i=1}^{N} p(y_i|\theta)\, p(\theta)$$

For Gaussian process, this becomes

$$Closed\ form\ GP = Gaussian\ data * GP\ prior$$

When we fit the model, we are learning the kernel and mean hyperparameters

Predictions are done using Bayes' formula as well

Posterior predictive distribution

$$p(y^{new}|y) \propto \int p(y^{new}|\theta)\, p(\theta|y) d\theta$$

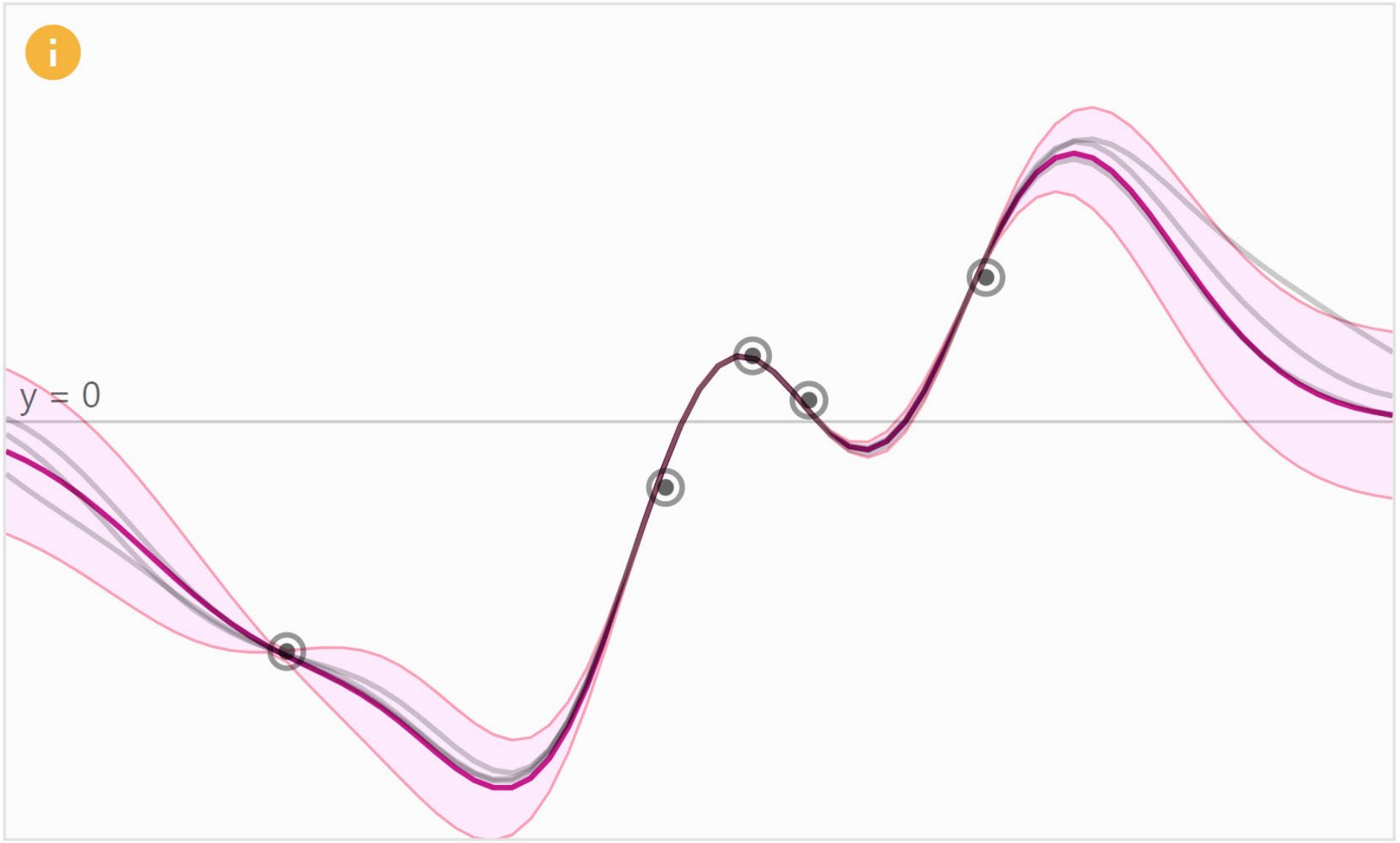
Likelihood is aleatoric uncertainty
Or noise in data itself

Posterior is epistemic uncertainty
Or our unknowns

Predictions are simply:

$$p(y^{new}|x^{new}, y, x) = N(\mu^{new}, \Sigma^{new})$$

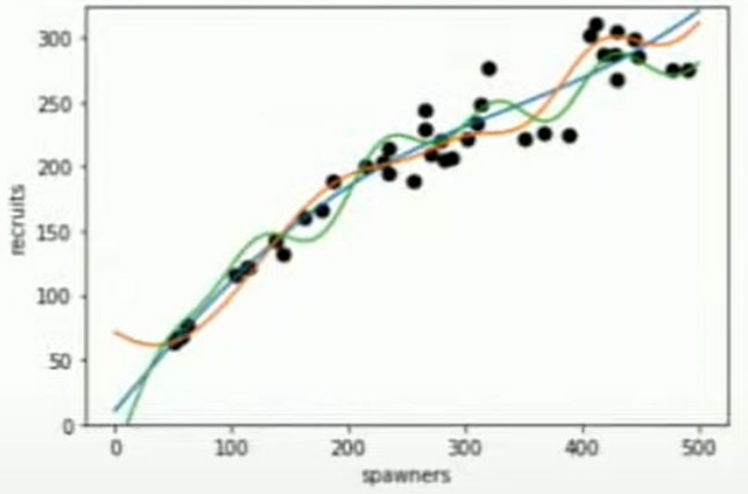
Confidence intervals and predictions come from the distribution statistics μ, Σ



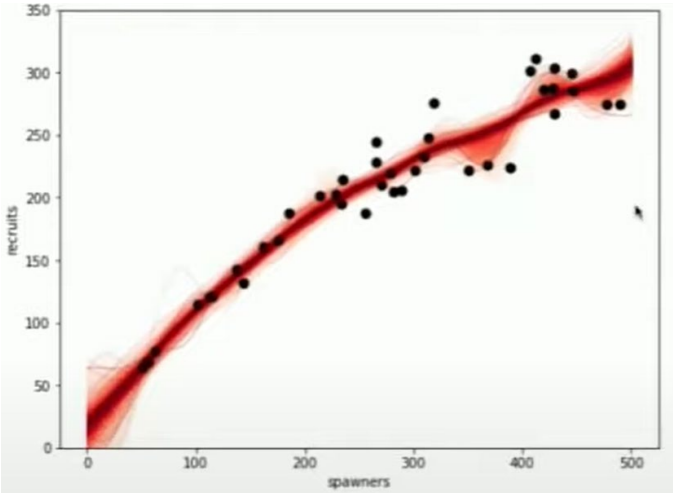
Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

Confidence intervals and predictions come from the distribution statistics μ, Σ

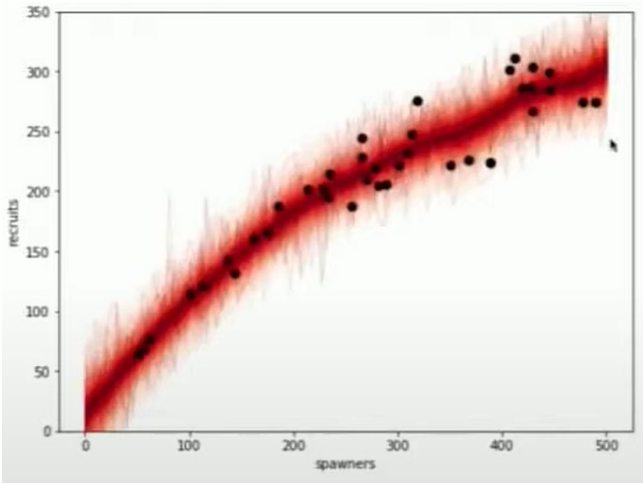
3 attempts at posteriors



100 attempts at posteriors



With prediction noise



Coding Tech (YouTube) A Primer on Gaussian Processes for Regression Analysis || Chris Fonnesbeck

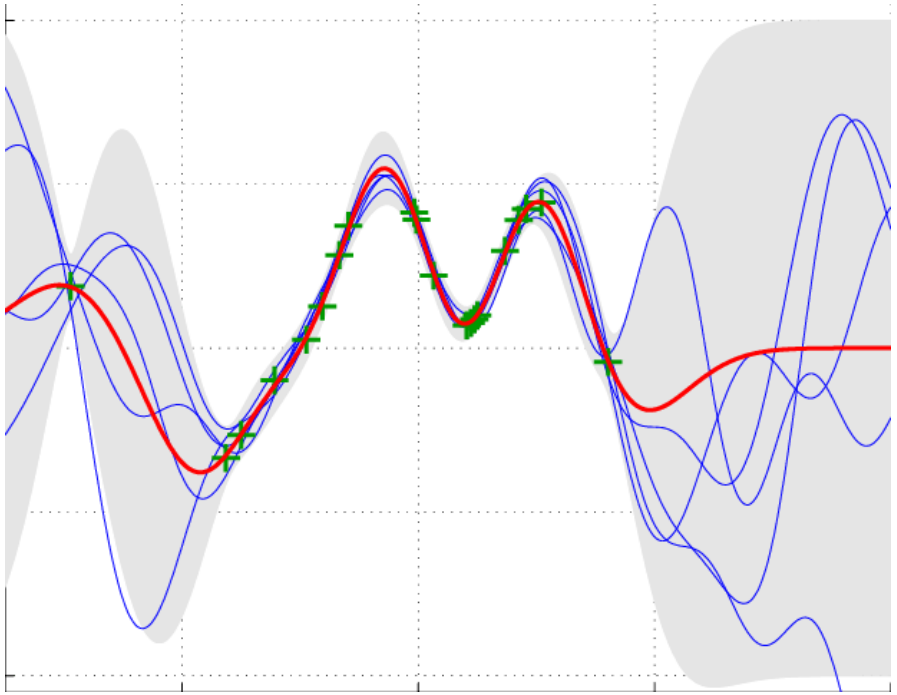
Limitations of Gaussian Processes

Extrapolation does really poorly with Gaussian processes (depends on length scale of kernel)

They are not sparse (they use the whole samples/features information to perform the prediction

They lose efficiency in high-dimensional spaces (features > a few dozen)

Picking and tuning kernel is tricky



Perez-Cruz et al Gaussian Processes for Nonlinear Signal Processing: An Overview of Recent Advances IEEE