Before we wrote that our experimental observation is a sum of sim, discrepancy, and measurement error

$$Y_{obs}(\vec{x}) = Y_{sim}(\vec{x}, \theta) + \delta(\vec{x}) + \varepsilon$$

- Y_{sim} might be expensive to evaluate, a problem for calibrating θ's
- We can build an emulator of the simulation by running Y_{sim} at several $\vec{x}$'s and θ's and use a regression model to give output between the points Y_{sim} was evaluated
- Then we also build a regression model for δ
- The most common way to do this for predictive models is through Gaussian Process Regression
- This is a nonparametric regression model, meaning we don't specify a functional form for the regression
- To demonstrate ~~the~~ GPR methodology we begin with a parametric example before invoking the non-parametric case

GP for Standard Linear Model

- We have a set D of data, n observations

$$D = \{(\vec{x}_i, y_i) \mid i = 1..n\}$$

This is called training data

- We can write this as a vector of length n $\vec{y}$ and $d \times n$ matrix $\underline{\underline{X}}$, $D = (\underline{\underline{X}}, \vec{y})$

- write a standard linear model with gaussian noise

$$y = \vec{X}^T\vec{w} + \varepsilon, \qquad \varepsilon \sim N(0, \sigma_n^2)$$

- ε is independent, identically distributed
- Therefore, the likelihood of n observations $\vec{y}$ given $\underline{\underline{X}}$ and $\vec{w}$ is

$$P(\vec{y}|\underline{\underline{X}}, \vec{w}) = \prod_{i=1}^{n} P(y_i|\vec{x}_i, \vec{w}) = \prod_{i=1}^{n} \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left[-\frac{(y_i - \vec{x}_i^T\vec{w})^2}{2\sigma_n^2}\right]$$

$$= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\vec{y} - \underline{\underline{X}}^T\vec{w}|^2\right) = N(\underline{\underline{X}}^T\vec{w}, \sigma_n^2 I)$$

each observation is independent

- Now we can use Bayesian formalism to get the weights
- Give a prior for $\vec{w} \sim N(\vec{0}, \Sigma_p) = \pi(\vec{w})$
- The posterior for $\vec{w}$ given D is

$$P(\vec{w}|\underline{\underline{X}}, \vec{y}) = \frac{P(\vec{y}|\underline{\underline{X}}, \vec{w})\,\pi(\vec{w})}{\int \pi(\vec{w})\,P(\vec{y}|\underline{\underline{X}}, \vec{w})\,d\vec{w}}$$

$$\propto \exp\left(-\frac{1}{2\sigma_n^2}(\vec{y} - \underline{\underline{X}}^T\vec{w})^T(\vec{y} - \underline{\underline{X}}^T\vec{w})\right) \exp\left(-\frac{1}{2}\vec{w}^T\Sigma_p^{-1}\vec{w}\right)$$

$$\propto \exp\left(-\frac{1}{2}(\vec{w} - \vec{w}^*)^T\left(\frac{1}{\sigma_n^2}\underline{\underline{X}}\underline{\underline{X}}^T + \Sigma_p^{-1}\right)(\vec{w} - \vec{w}^*)\right)$$

$$\vec{w}^* = \sigma_n^{-2}\left(\sigma_n^{-2}\underline{\underline{X}}\underline{\underline{X}}^T + \Sigma_p^{-1}\right)^{-1}\underline{\underline{X}}\vec{y}$$

$$= \frac{1}{\sigma_n^2}A^{-1}\underline{\underline{X}}\vec{y}$$

- The posterior is the gaussian with mean $\vec{w}^*$ and covariance A^{-1}

$$p(\vec{w}|\underline{\underline{X}}, y) \sim N(\vec{w}^*, A^{-1})$$

- Now we can make predictions at test points $\vec{x}^*$ by averaging over all possible linear models w.r.t. to Gaussian posterior

$$p(\vec{x}^{*T}\vec{w} \mid \vec{x}^*, \underline{X}, \vec{y}) = \int p(\vec{x}^{*T}\vec{w} \mid \vec{x}^*, \vec{w})\, p(\vec{w} \mid \underline{X}, \vec{y})\, d\vec{w}$$

$$= N\left(\frac{1}{\sigma_n^2}\vec{x}^{*T} A^{-1} \underline{X} \vec{y},\ \vec{x}^{*T} A^{-1} \vec{x}^*\right)$$

- the pred. variance is quadratic in test input so uncertainties grow with x^*

Example

$y = w_1 + w_2 x_1 + \varepsilon$

$\vec{w} \sim N(\vec{0}, I)$ for posterior, $\varepsilon \sim N(0, 1)$

We begin with design points

$X_1 = -5, 1, 5$

$y = -5.1, 0.25, 4.9$

(Shows figs)

Now add two points

and then change σ_n^2 from 1 to 0.1

Generalization to other bases (feature space)

- Instead of a linear model, we use a general function space, $\phi(\vec{x})$
- Example if $\vec{x} = x_1$, $\phi(x_1)$ could be $\phi(x_1) = \{1, x_1, x_1^2, x_1^3, \ldots\}$
- Later we'll see that we don't have to pick $\phi(\vec{x})$
- $\phi(\vec{x}): \vec{x} \rightarrow f$ (length d, length N)
- Define matrix of $\phi(\vec{x})$ as $\underline{\Phi}(\underline{X})$: $n \times N$ matrix (n: number of test points)

- Now y has the form

$$y = \phi(\vec{x})^T \vec{w} + \varepsilon$$

- And the analysis is analagous to the linear case and we get predictive dist as

$$p(\phi(\vec{x})^T \vec{w} \mid \vec{x}^*, \underline{X}, \vec{y}) \sim N\left(\frac{1}{\sigma_n^2} \phi(\vec{x}^*)^T \underline{A}^{-1} \underline{\Phi}(\underline{X}) \vec{y},\ \phi(\vec{x}^*)^T \underline{A}^{-1} \phi(\vec{x}^*)\right)$$

$$\underline{A} = \sigma_n^{-2} \underline{\Phi}(\underline{X}) \underline{\Phi}(\underline{X})^T + \Sigma_p^{-1}$$

- This could be difficult to compute if $\underline{A}$ is large
- To get around this we use the Kernel trick
- The feature space always enters in the general form

$$k(x, x') = \phi(\vec{x})^T \Sigma_p \phi(\vec{x}')$$

if we rewrite

$$p(\phi(\vec{x}^*)^T \vec{w} \mid \vec{x}^*, \underline{X}, \vec{y}) \sim N\left(\phi(\vec{x}^*) \Sigma_p \underline{\Phi}(\underline{X}) (K + \sigma_n^2 I)^{-1} y,\ \right.$$

$$\left. \phi_*^T \Sigma_p \phi(\vec{x}^*) - \phi(\vec{x}^*)^T \Sigma_p \underline{\Phi}(\underline{X}) (K + \sigma_n^2 I)^{-1} \underline{\Phi}(\underline{X})^T \Sigma_p \phi(\vec{x}^*)\right)$$

$$K = \underline{\Phi}(\underline{X})^T \Sigma_p \underline{\Phi}(\underline{X})$$

- Now using $k(x, x')$ we can make the predictions defined only using weighted inner products.
- We can define different kernels to get different feature spaces
- Defining a Kernel is easier than defining a feature space

Now we will define Gaussian Process Regression as a space of functions with certain properties

Gaussian Process - A collection of random variables, any finite subset of which have a joint Gaussian distribution

- A Gaussian process is completely determined by its mean and covariance function.
- Given a real process $f(\vec{x})$ the mean function is

$$m(\vec{x}) = E[f(\vec{x})]$$

the covariance function is

$$k(\vec{x}, \vec{x}') = E\left[(f(\vec{x}) - m(\vec{x}))(f(\vec{x}') - m(\vec{x}'))\right]$$

- For GPR the random variables are the value of $f(\vec{x})$ at $\vec{x}$ they have a known mean and covariance
- The way a GP is defined, if $(y_1, y_2) \sim N(\vec{\mu}, \underline{\underline{\Sigma}})$ then $y_1 \sim N(\mu_1, \Sigma_{11})$
- We will write our Gaussian process as

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

- A common covariance function is the squared exponential function

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

- This cov function is equivalent to Bayesian regression model with an infinite number of basis functions
- This is an example of how it is easier to pick Kernel than specify $\phi(\vec{x})$

- once we have a covariance function we have a distribution over functions.
- To demonstrate this lets specify

$$f(\underline{\underline{X}}^*) \sim N(\vec{0}, K(\underline{\underline{X}}^*, \underline{\underline{X}}^*))$$

here $\underline{\underline{X}}$ is a matrix of input points $\vec{x}$,
K is a covariance matrix given by

$$K_{ij} = k(\vec{x}_i, \vec{x}_j)$$

- Lets specify $\underline{\underline{X}}^* = [-10, -9.9, \ldots, 9.9, 10]$ and use SE cov function, $\ell = 1$
- Then by sampling from $N(\vec{0}, K(\underline{\underline{X}}^*, \underline{\underline{X}}^*))$ we get random functions evaluated at $\underline{\underline{X}}^*$

Prediction from noise-free observations

- We can now generate functions from distributions
- If we consider $f(\underline{\underline{X}}^*) \sim N(\vec{0}, K(\underline{\underline{X}}^*, \underline{\underline{X}}^*))$ a prior and we have a set of training points

$$\{(x_i, f_i) \mid i = 1..n\}$$

We have a prior distribution

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} \sim N\left(\vec{0}, \begin{bmatrix} K(\underline{\underline{X}}, \underline{\underline{X}}) & K(\underline{\underline{X}}, \underline{\underline{X}}^*) \\ K(\underline{\underline{X}}^*, \underline{\underline{X}}) & K(\underline{\underline{X}}^*, \underline{\underline{X}}^*) \end{bmatrix}\right)$$

- Where for n^* prediction points, $K(\underline{\underline{X}}^*, \underline{\underline{X}}^*)$ is $n^* \times n^*$ and $K(\underline{\underline{X}}, \underline{\underline{X}}^*)$ is $n \times n^*$

- What we want to do is restrict this distribution so that fonctions that don't agree with observations $\vec{f}$ are rejected.
- Because $\vec{f}, \vec{f}^*$ are jointly Gaussian (by def. of GP)

we can use the property that for jointly Gaussian random vects

$$\begin{bmatrix} \vec{X} \\ \vec{Y} \end{bmatrix} \sim N\left(\begin{bmatrix} \vec{\mu}_X \\ \vec{\mu}_Y \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right)$$

$$p(\vec{X}|\vec{Y}) \sim N(\vec{\mu}_X + CB^{-1}(\vec{Y} - \vec{\mu}_Y),\ A - CB^{-1}C^T)$$

to write

$$p(\vec{f}^*|\vec{f}, \underline{\underline{X}}^*, \underline{\underline{X}}) \sim N(K(\underline{\underline{X}}_*, \underline{\underline{X}})K(\underline{\underline{X}}, \underline{\underline{X}})^{-1}\vec{f},$$
$$K(\underline{\underline{X}}^*, \underline{\underline{X}}^*) - K(\underline{\underline{X}}^*, \underline{\underline{X}})K(\underline{\underline{X}}, \underline{\underline{X}})^{-1}K(\underline{\underline{X}}, \underline{\underline{X}}^*))$$

- Now we simply need to compute $K(\underline{\underline{X}}^*, \underline{\underline{X}})$, $K(\underline{\underline{X}}^*, \underline{\underline{X}}^*)$, $K(\underline{\underline{X}}, \underline{\underline{X}})^{-1}$

Example: Take 5_{train} points and constrain previous example.

- the data interpolates points, away from training points uncertainty goes up.

Prediction with noisy observations

- In this case we don't know $f(\vec{x})$, rather $y = f(\vec{x}) + \varepsilon$

now write

$$\text{Cov}(\vec{y}) = K(\underline{\underline{X}}, \underline{\underline{X}}) + \sigma_n^2 \underline{\underline{I}}$$

- The joint prior is

$$\begin{bmatrix} \vec{Y} \\ \vec{f}^* \end{bmatrix} \sim N\left(\vec{0}, \begin{bmatrix} K(\underline{X}, \underline{X}) + \sigma_n^2 \underline{\underline{I}} & K(\underline{X}, \underline{X}^*) \\ K(\underline{X}^*, \underline{X}) & K(\underline{X}^*, \underline{X}^*) \end{bmatrix} \right)$$

- The posterior distribution is

$$p(\vec{f}^* \mid \underline{X}, \vec{Y}, \underline{X}^*) \sim N(\hat{f}^*, \text{cov}(\vec{f}^*))$$

where $$\hat{f}^* = K(\underline{X}^*, \underline{X}) \left[K(\underline{X}, \underline{X}) + \sigma_n^2 \underline{\underline{I}} \right]^{-1} \vec{Y}$$

$$\text{cov}(\vec{f}^*) = K(\underline{X}^*, \underline{X}^*) - K(\underline{X}^*, \underline{X}) \left[K(\underline{X}, \underline{X}) + \sigma_n^2 \underline{\underline{I}} \right]^{-1} K(\underline{X}, \underline{X}^*)$$

What do the covariance params mean

$$K(\vec{X}_a, \vec{X}_b) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2} |\vec{X}_a - \vec{X}_b|^2 \right) + \sigma_n^2 \delta_{ab}$$

ℓ = length scale, σ_f^2 = signal variance, σ_n^2 = noise variance

- ℓ controls how oscillatory the GP is
- if ℓ is too small function can interpolate data better, but less accurate between
- if ℓ is to large function might not bend enough to hit the obs.
- low σ_f means that data is ~~not~~ less significant
- high σ_f means that all data (both exp and sim) is sig.

Empirical Bayes Approach

- What if we have no idea how to pick $\ell, \sigma_f^2, \sigma_n^2$?
- We can use Bayesian inference to pick them
- To more closely match the notation in the literature we'll write our covariance function

$$k(\vec{x}, \vec{x}') = \frac{1}{\lambda_n} \exp\left[-\sum_{k=1}^{n} \beta_k |x_k - x_k'|^\alpha\right]$$

$$\lambda_n = \frac{1}{\sigma_f^2} \qquad \beta_k = \frac{1}{2\ell_k}$$

- Now we have different ℓ's for different elements of $\vec{x}$.
- Given $\vec{z} = \begin{bmatrix} \vec{y} \\ \vec{f}^* \end{bmatrix}$ what is the likelihood of getting the output $\vec{z}$ with ~~a~~ GP with cov. params λ_n
- we also now allow general mean, μ
- The likelihood is

$$P(\vec{z} | \mu, \lambda_n, \vec{\beta}) \propto |\Sigma_z|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\vec{z} - \mu\vec{1})^T \Sigma_z^{-1} (\vec{z} - \mu\vec{1})\right]$$

$$\Sigma_z = \begin{bmatrix} K(\underline{\underline{X}}, \underline{\underline{X}}) + \sigma_n^2 \underline{\underline{I}}, & K(\underline{\underline{X}}, \underline{\underline{X}}') \\ K(\underline{\underline{X}}', \underline{\underline{X}}) & K(\underline{\underline{X}}', \underline{\underline{X}}') \end{bmatrix}$$

- Therefore, the posterior distribution of the parameters is

$$P(\mu, \lambda_n, \vec{\beta} | \vec{z}) \propto p(\vec{z} | \mu, \lambda_n, \vec{\beta}) \pi(\vec{\beta}) \pi(\mu) \pi(\lambda_n)$$

- Common choices of priors

$$\pi(\mu) \propto \exp\left(-\frac{\mu^2}{2v}\right) \quad (v=0 \Rightarrow \mu=0)$$

$$\pi(\lambda_n) \propto \lambda_n^{\alpha-1} \exp(-b\lambda_n), \quad \lambda_n > 0, \quad (a, b \text{ large encourages } \lambda_n \text{ to be near})$$

$$\pi(\vec{\beta}) \propto \prod_{k=1}^{n} \left(1 - e^{-\beta_k}\right)^{-\frac{1}{2}} e^{-\beta_k} \qquad \beta_k > 0 \text{ (encourages large } \beta_k, \text{ small } \theta)$$

- Now we can use MCMC to find did. params
- This will be inexpensive because we don't need more code runs, the runs are already contained in $\vec{y}$

Adding Discrepancy Term

Recall discrepancy term:

$$y_{obs} = y_{sim}(\vec{x}, \theta) + \delta(\vec{x}) + \varepsilon$$

- We use a Gaussian process to model δ, with mean zero and covariance

$$k(\vec{x}, \vec{x}') = \frac{1}{\lambda_\delta} \exp\left(-\sum_{k=1}^{n} \beta_k^\delta |x_k - x_k'|^{\alpha_\delta}\right)$$

- The likelihood has the same form

$$p(\vec{z} | \mu, \lambda_z, \vec{\beta}, \lambda_\delta \vec{\beta}^\delta) \propto |\Sigma_z|^{-1/2} \exp\left[-\frac{1}{2}(\vec{z} - \mu\vec{1})^T \Sigma_z^{-1} (\vec{z} - \mu\vec{1})\right]$$

where now

$$\Sigma_z = \begin{bmatrix} K(\underline{x}, \underline{x}) + \sigma_n^2 \underline{\underline{I}} + K_\delta(\underline{x}, \underline{x}) & K(\underline{x}, \underline{x}') \\ K(\underline{x}', \underline{x}) & K(\underline{x}', \underline{x}') \end{bmatrix}$$