Gaussian Processes Regression

- Similar to Bayesian regression, but we get a posterior on the actual function and not the parameters.
- GP can also be used for classification, but not as popular

GP: let $f: \mathcal{X} \to \mathbb{R}$, a GP is a collection of RVs $\{f(x) : x \in \mathcal{X}\}$ s.t. any finite collection, $x_1, x_2, \ldots x_m \in \mathcal{X}$ is MVN:

$$\begin{bmatrix} f(x^{(1)}) \\ \vdots \\ f(x^{(m)}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x^{(1)}) \\ \vdots \\ m(x^{(m)}) \end{bmatrix}, K \right) \quad \text{s.t. } K \text{ is PSD}$$

K can be written as

$$K = \begin{bmatrix} k(x^{(1)}, x^{(1)}), & k(x^{(1)}, x^{(2)}) & \cdots \\ \vdots & \ddots & \\ k(x^{(m)}, x^{(1)}) & \cdots & k(x^{(m)}, x^{(m)}) \end{bmatrix}; \quad m: \mathbb{R}^n \to \mathbb{R}, \; k: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

$\Rightarrow$ i.e., k is a valid kernel

(k quantifies the autocorrelation of the GP regression)

- i.e., a GP puts a probability dist. over functions
- For a GP, we say that $f \sim GP(m, k)$

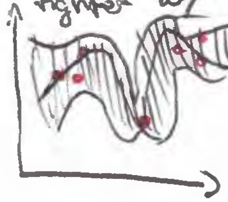
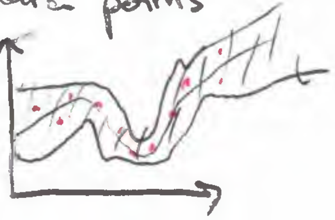
GP regression

model: $y^{(i)} = f(x^{(i)}) + \varepsilon^{(i)}$ where $f \sim GP(m, K)$, $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$

$\Rightarrow$ [illegible]

$$Y \sim \mathcal{N}(m, \underline{K} + \sigma^2 I_m)$$

- intervals get "pinched" around observations and get tighter w/ more data points

- using some nice formulas which say that the marginal and conditional dists of MVNs are MVNs, it is not difficult to show that for new points $\{y^*, x^*\}$ the posterior $P(y^* | x^*, y, X)$ is MVN w/ a mean and covariance matrix that can be computed w/ fast linear algebra:

$$Y^* | Y, X^*, X \sim \mathcal{N}(\mu^*, K^*)$$

$\Rightarrow$ thus, we can predict the maximum probability of y^* (i.e., μ^*) and we also have the full posterior to quantify our uncertainty.

- hyperparameter tuning: usually done by maximizing the marginal likelihood $P(y | X; \Theta)$

Pros

- get full posterior
  - can quantify uncertainty well
  - can use for decision theoretic purposes (eg. Bayesian optimization)
- non-parametric
- highly flexible (using different kernels)

~~[illegible]~~

Cons

- does not work well w/ many dims (because you need to fit a full distribution)
- computation time is high, due to inverting matrices, $O(n^3)$, so you can only use w/ a few thousand data points