1、Definition of stochastic processes: (Kolmogoro Extension Theorem)

Given a random function $F: \mathcal{X} \to \mathcal{Y}$

For any finite sequence $x_{1:n} = (x_1, \cdots, x_n)$, $\forall n \in N$

$$y_{1:n} = (F(x_1), \cdots, F(x_n))$$

① Exchangeability: for any permutation $\pi(x_{1:n})$ and $\pi(y_{1:n})$

(permutation invariance) $P_{x_{1:n}}(y_{1:n}) = P_{\pi(x_{1:n})}(\pi(y_{1:n}))$ ← Marginals remain as the same

② Consistency: for $1 \le m \le n$

$$P_{x_{1:m}}(y_{1:m}) = \int P_{x_{1:n}}(y_{1:n})\, dy_{m+1:n}$$

An instantiation: $P_{x_{1:n}}(y_{1:n}) = \int p(f)\, p(y_{1:n} | f, x_{1:n})\, df$ (1)

more specifically: $P_{x_{1:n}}(y_{1:n}) = \int p(f) \prod_{i=1}^{n} N(y_i / f(x_i), \sigma^2)\, df$

Gaussian Process (GP): $f \sim GP(\mu(x), k(x, x'))$

2、Neural Processes:

Introduce an auxiliary variable z:

Rewrite $F(x) = g(x, z)$, g is a NN, g is stochastic NN

We get a generative model: (latent z)

$$p(z, y_{1:n} | x_{1:n}) = p(z) \prod_{i=1}^{n} N(y_i / g(x_i, z), \sigma^2) \quad (2)$$

yields $p(z | x_{1:n}, y_{1:n})$

3、Variational Auto-Encoder:

$$\log p(y_{1:n} | x_{1:n}) \ge E_{q(z|x_{1:n}, y_{1:n})}\left[\sum_{i=1}^{n} \log p(y_i / z, x_i) + \log \frac{p(z)}{q(z | x_{1:n}, y_{1:n})} \right]$$

log evidence

Encoder $q(z \mid x_{1:n}, y_{1:n})$ $\quad \begin{matrix} x_{1:n} \\ y_{1:n} \end{matrix} > \begin{matrix} \mu_z \\ \sigma_z^2 \end{matrix} \to z$

Decoder $P(y_i \mid z, x_i)$

Spilt data set: Context set $\{x_{1:m}, y_{1:m}\}$

target set $\{x_{m+1:n}, y_{m+1:n}\}$

$$\log P(y_{m+1:n} \mid x_{1:n}, y_{1:m}) \geq$$

$$E_{q(z \mid x_{1:n}, y_{1:n})}\left[\sum_{i=m+1}^{n} \log P(y_i \mid z, x_i) + \log \frac{P(z \mid x_{1:m}, y_{1:m})}{q(z \mid x_{1:n}, y_{1:n})}\right]$$

Encoder $q(z \mid x_{1:n}, y_{1:n})$, using both Context and target sets

Decoder $P(y_i \mid z, x_i)$, only decodes target set

Replace $P(z \mid x_{1:m}, y_{1:m})$ with a data driven prior

$q(z \mid x_{1:m}, y_{1:m})$

At testing, all observations $\{x_{1:n}, y_{1:n}\}$ are utilized to construct $q(z \mid x_{1:n}, y_{1:n})$

Make prediction at x^* by

$$P(y^* \mid x^*, y_{1:n}, x_{1:n}) = \int q(z \mid x_{1:n}, y_{1:n}) \cdot N(y^* \mid g(x^*, z), \sigma^2) dz$$

Meta learning: learn task level information

amplitude

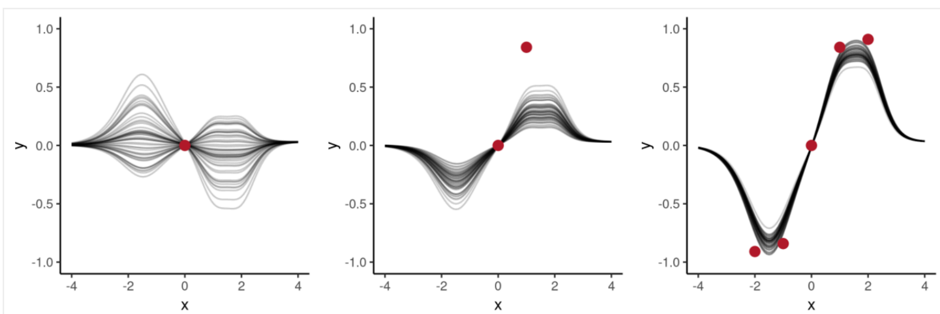
sinusoid Regression: $y = f(x) = a\sin(x)$

a function class $\{f \in \mathcal{M}\}$

For example: 1. $a \sim U[-2, 2]$

2. sample (x_i, y_i)s from curve $y = a\sin(x)$

3. at each training iteration, spilit (x_i, y_i)s into context and target sets to do optimization



$a = 1$