Neural Processes

Alexander Plakhin, Nikolai Averianov

HSE FES Probability Theory Club

November 11, 2023

Motivation

- Gaussian Processes Regression provides a probabilistic framework to impose prior distribution over a set of functions and update this distribution after observing the data
- Obvious drawbacks are computational complexity ($\mathcal{O}(N^3)$) and limited expressiveness
- Neural Processes is a similar framework which addresses the issues above

Gaussian Processes Regression (Reminder)

- We assume that you know what is a Gaussian Processes Regression

Stochastic processes

- Let's consider the process $F : \mathcal{X} \to \mathcal{Y}$ as a random function. We define distribution over $Y_{1:n} := (F(x_1), \ldots, F(x_n))$
- Imagine we have a collection of distributions $\rho_{x_{1:n}}$. What are the conditions for this collection to define a stochatic process F?
- Exchageability (for any permutation):

$$\rho_{x_1,\ldots,x_n}(y_1, \ldots, y_n) = \rho_{x_{\pi(1)},\ldots,x_{\pi(n)}}(y_{\pi(1)}, \ldots y_{\pi(n)})$$

- Consistency:

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

- These conditions are sufficient by the Kolmogorov Extension Theorem
- We need to build a Neural Network that satisfies the conditions above

Model Setup

- Given a particular instantiation of the stochastic process f:

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

- if we add noise: $Y_i \sim \mathcal{N}(F(x_i), \sigma^2)$:

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

- Now we can do the following: let's assume that realization of the process depends on a global latent variable z and parameterize $F(x)$ as a Neural Network $g_\theta(x, z)$:

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

ELBO

- Let q be variational posterior parametrized by another NN. ELBO (expectations are w.r.t $q(z|x_{1:n}, y_{1:n})$):

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

- But for this problem we really care more about conditional sampling. So we have:

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

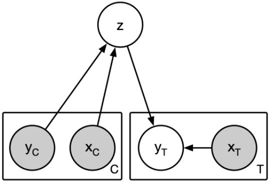
- Of course we have no clue about $p(z|x_{1:n}, y_{1:m}) \Rightarrow$ we end up maximizing:

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

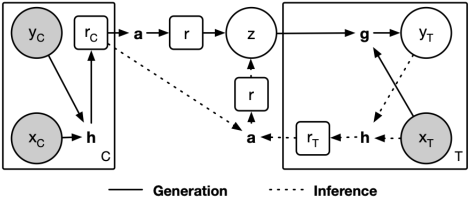
Architecture

The model consist of several parts:

- Encoder h that takes pairs $(x, y)_i$ and maps them to representation r_i
- Aggregator a that summarizes inputs and is order-invariant. We need a single global representation $r = a(r_1, \ldots, r_n)$ to parameterize $z \sim \mathcal{N}(\mu(r), \sigma^2(r)I)$. Note linear complexity of aggregator in case $a(r_1, \ldots, r_n) = \sum_{i=1}^{n} r_i$
- Conditional decoder g that takes new points x and the sampled latent variable z

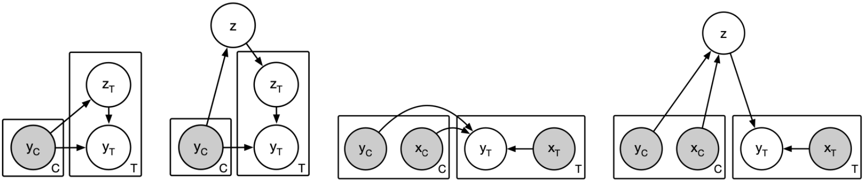


(a) Graphical model



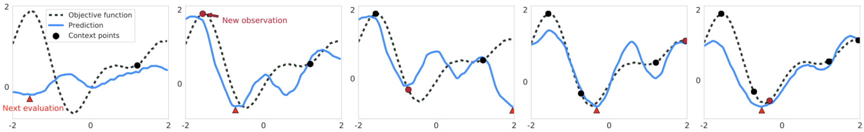
(b) Computational diagram

Similar models

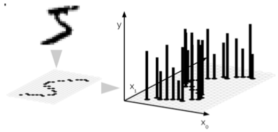


(a) Conditional VAE (b) Neural statistician (c) Conditional neural process (d) Neural process

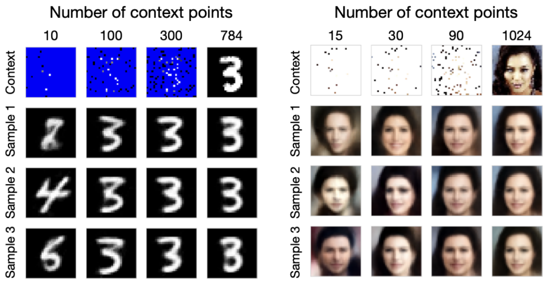
Results



Objective function
Prediction
Context points
New observation
Next evaluation



y
x1
x0



Number of context points
10 100 300 784
15 30 90 1024
Context
Sample 1
Sample 2
Sample 3

Sources

1. Neural Processes – https://arxiv.org/pdf/1807.01622.pdf
2. https://yanndubs.github.io/Neural-Process-Family/text/Intro.html
3. https://github.com/EmilienDupont/neural-processes