Gaussian Processes (Some Other Aspects)

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Scalability of GPs

- Computational costs in some steps of GP models scale in the size of training data
- For example, prediction cost is $O(N)$

$O(N)$ cost assuming $\mathbf{C}_N$ is already inverted

$$p(y_*|\boldsymbol{y}) = \mathcal{N}(\mu_*, \sigma_*^2) \qquad \mu_* = \mathbf{k}_*^\top \mathbf{C}_N^{-1}\boldsymbol{y} \qquad \sigma_*^2 = \kappa(x_*, x_*) - \mathbf{k}_*^\top \mathbf{C}_N^{-1}\mathbf{k}_* + \beta^{-1}$$

- GP models often require matrix inversions (e.g., in marg-lik computation when estimating hyperparameters) – takes $O(N^3)$
- Storage also requires $O(N^2)$ since need to store the covariance matrix
- A lot of work on speeding up GPs[1]. Some prominent approaches include
  - Inducing Point Methods (condition predictions only on a small set of "learnable" points)
  - Divide-and-Conquer (learn GP on small subsets of data and aggregate predictions)
  - Kernel approximations

$M \ll N$ pseudo-inputs and pseudo-outputs

- Note that nearest neighbor methods and kernel methods also face similar issues
  - Many tricks to speed up kernel methods can be used for speeding up GPs too

[1]When Gaussian Process Meets Big Data: A Review of Scalable GPs - Liu et al, 2018

GP: Some Comments

- GP is sometimes referred to as a nonparametric model because
  - Complexity (representation size) of the function f grows in the size of training data
  - To see this, note the form of the GP predictions, e.g., predictive mean in GP regression

$$\mu_* = f(\mathbf{x}_*) = \mathbf{k}_*{}^\top \mathbf{C}_N^{-1} \mathbf{y} = \mathbf{k}_*{}^\top \boldsymbol{\alpha} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_*, \mathbf{x}_n)$$

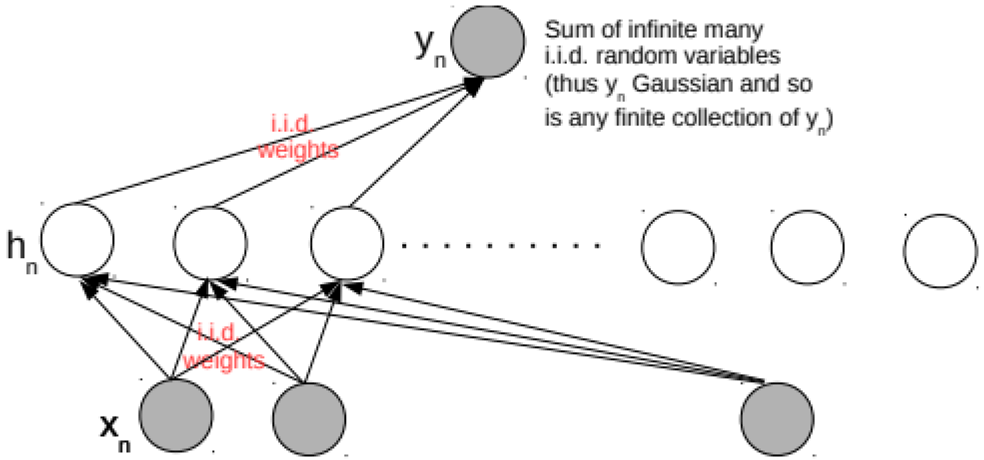
  - It implies that $f(.) = \sum_{n=1}^{N} \alpha_n k(., \mathbf{x}_n)$ which means f is written in terms of all training examples
  - Thus the representation size of f depends on the number of training examples

- In contrast, a parametric model has a size that doesn't grow with training data
  - E.g., a linear model learns a weight vector $\boldsymbol{w} \in \mathbb{R}^D$ (D parameters, size independent of N)

- Nonparametric models more flexible since their complexity is not limited beforehand
  - Note: Methods like nearest neighbors and kernel SVMs are also nonparametric (but not Bayesian)

Neural Networks and Gaussian Process

- An infinitely-wide single hidden layer NN with i.i.d. priors on weights = GP
- Shown formally by (Neal[2], 1994). Based on applying the central limit theorem



- This equivalence is useful for several reasons
  - Can use a GP instead of an infinitely wide Bayesian NN (which is impractical anyway)
  - With GPs, inference is easy (at least for regression and with known hyperparams)
  - A proof that GPs can also learn any function (just like infinitely wide neural nets - Hornik's theorem)
- Connection generalized to infinitely wide multiple hidden layer NN (Lee et al[3], 2018)

[2]Priors for infinite networks, Tech Report, 1994
[3]Deep Neural Networks as Gaussian Processes (ICLR 2018)

GP: A Few Other Comments

- GPs can be thought of as Bayesian analogues of kernel methods (like RVMs)
  - Can get estimate in the uncertainty in the function and its predictions

  - Can learn the kernel (by learning the hyperparameters of the kernels)
- Not limited to supervised learning problems
  - f could even define a mapping of low-dim latent variable z_n to an observation x_n

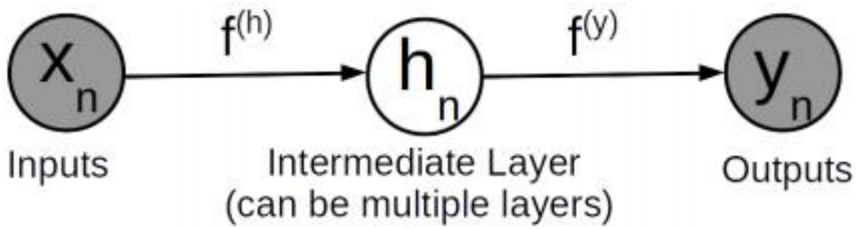
$$x_n = f(z_n) + \text{"noise"}$$

GP latent variable model for dimensionality reduction (like a kernel version of probabilistic PCA)

- Many mature implementations of GP exist. You may check out
  - GPyTorch (PyTorch), GPFlow (Tensorflow)
  - GPML (MATLAB), GPsuff (MATLAB/Octave)

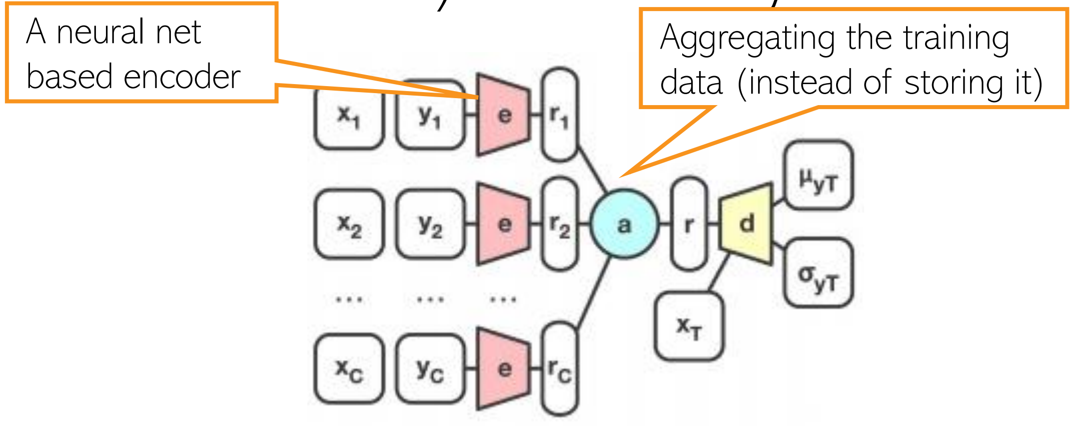
GP: Some Other Recent Advances

- Deep Gaussian Processes (DGP)
  - Akin to a deep neural network where each hidden node is modeled by a GP



  - A nice alternative to linear transform + nonlinearity in neural nets, e.g., $\boldsymbol{h} = \tanh(\boldsymbol{W}\boldsymbol{x})$

- GPs with deep kernels defined by neural nets
- Neural Processes (GP + neural nets): Faster way to do GPs



Coming Up

- Sequential decision making using Bayesian methods
  - Active Learning
  - Bayesian Optimization