Task-Agnostic Amortized Inference of Gaussian Process Hyperparameters

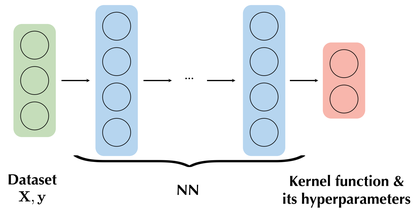
Sulin Liu, Xingyuan Sun, Peter J. Ramadge, Ryan P. Adams
Princeton University

PRINCETON LABORATORY FOR INTELLIGENT PROBABILISTIC SYSTEMS

Introduction

Model selection problem in Gaussian processes: Choose the kernel function that accurately reflects the properties of the data.

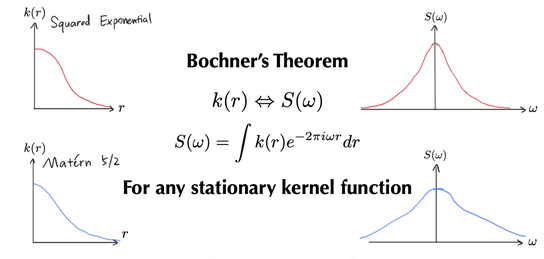
- **Issues** with (the most commonly used) marginal likelihood maximization:
  1. Tricky: What kernel function to use? Squared Exponential or Matérn? Periodicity?
  2. Costly: $\mathcal{O}(N^3)$ per optimization iteration
  3. Suboptimality: non-concave optimization
- **Idea**: Can we achieve tricky automatic and costly lightweight kernel hyperparameters identification via "*amortized*" inference?



- **Answer**: **Yes!** With a single neural net trained with only **synthetic data** and for different tasks with different input dimensionality.
- We propose amortized hyperparameter inference for Gaussian processes (AHGP), which speeds up GP hyperparameter inference by $\sim$**100**$\times$ on average while remaining comparable in quality.
- **Try our model at**: `https://github.com/PrincetonLIPS/AHGP`

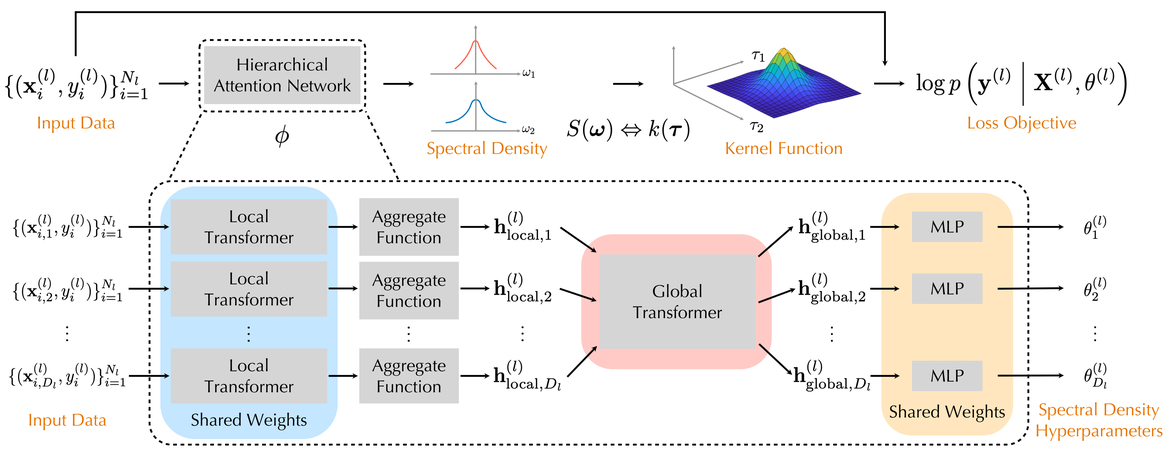
Modeling the Kernel Functions

- **Compositional modeling ?** (Duvenaud et al. [1]): Model a kernel by a compositions of base kernels. Kernel types, kernel hyperparameters and composition rules are modelled as latent random variables to be inferenced.
- **Challenges**:
  1. Huge search space that involves a mix of discrete and continuous variables
  2. Some variables are highly intercorrelated (such as lengthscale and kernel type)
- **Spectral modelling** ✓ (Wilson and Adams [3]):



- Predict the **spectral density** instead!
  A *unified*, *compact* and *continous* space for modeling the kernel functions.
  - Use Gaussian mixtures for modelling the spectral density: 1. universal approximator of densities and 2. closed form kernel function.
  - Assume product structure in the kernel matrix: for interpretability and flexibility

Amortized Hyperparameter Inference for Gaussian Process (AHGP)



The top part of the figure gives an illustration of the computation graph in AHGP. The bottom part describes our hierarchical attention neural net architecture.

Hierarchical Attention Network

- **Input**: a set of data points, $\{\mathbf{x}_i, y_i\}_{i=1}^N := \{\mathbf{X}, \mathbf{y}\}$
- **Output**: spectral density hyperparamters, θ
- What properties do we want?
  - *Versatility*: able to take in dataset of arbitrary data cardinality and dimensionality
  - *Permutation invariance/equivariance*: with respect to datapoint order and dimension order

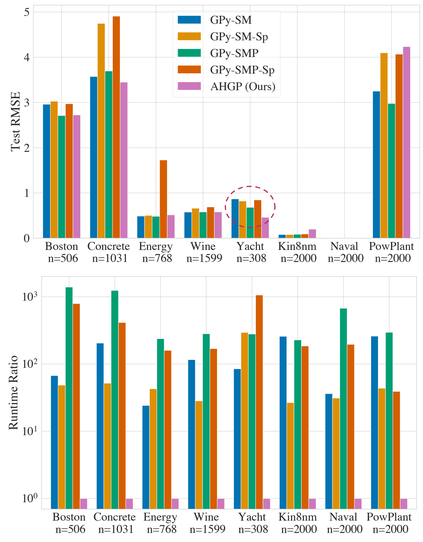
  So that a single trained neural model can be used for different tasks.
- Hierarchical attention architecture:
  - *Local Transformer*: encodes local per dimensional information about the function
  - *Aggegrate Function*: aggregates local per diemensional representations to a single vector
  - *Global Transformer*: models interactions between dimensions and encodes context-aware representations at a dimension level
- **Proposition:** *If AggrFunc is permutation invariant, LocalTransformer and MLP are weight-tied across different dimensions, then the hierarchical attention network is permutation invariant/equivariant.*
- **Complexity**: $\mathcal{O}(N^2 + D^2)$, where N is the number of data points and D is the dimensionality.
  - Note: possible to reduce to $\mathcal{O}(N + D)$ by using sparse or inducing point attentions.

Experimental Setup

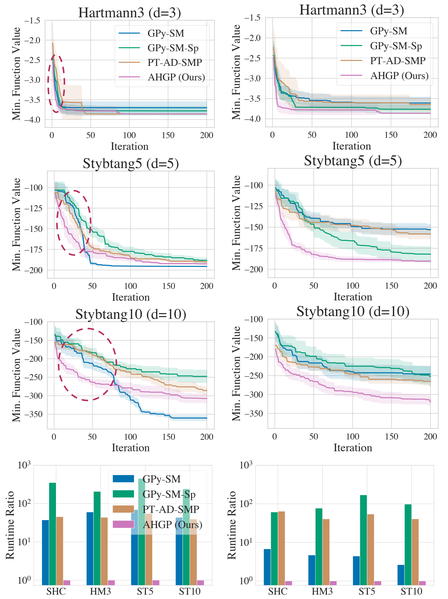
- **Training data**: 5K *synthetic datasets* generated from GP priors with stationary kernels of input dimensions 2$\sim$15, $\sim$30 data points
- **Training objective**: averaged marginal likelihood
  $$\mathcal{L}\left(\phi, \{\mathcal{D}^{(l)}\}_{l=1}^L\right) = -\frac{1}{L}\sum_{l=1}^L \frac{1}{N_l} \log p\left(\mathbf{y}^{(l)} \mid \mathbf{X}^{(l)}, \theta^{(l)}\right)$$
- **Training method**: Adam with fixed learning rate, dropout on self-attention encoders
- **Architecture**: 8 layers for both Local and Global Transformer, self-attention hidden dimension 256 with 4 heads.
- **Testing**: The *same trained neural model* will be used to predict kernel hyperparameters for all *unseen* test cases.
- **Baselines**:
  - Method: marginal likelihood maximization for full GP and sparse variational GP (Titsias [2])
  - Kernel: spectral mixture kernel (SM) and spectral mixture product kernel (SMP)
  - Package: GPy on CPU and GPyTorch on GPU

Experimental Results

- UCI regression benchmarks: 90%-10% train-test split



- Bayesian optimization benchmarks: expected improvement (EI) is used as the acquisition function



Left: Baseline methods use random re-initialization strategy.
Right: Baseline methods use warmstart initialization strategy.

- AHGP produces comparable-quality GP kernel hyperparameters while being $\sim$100$\times$ faster.
- In addition, AHGP demonstrates robustness in the low-data regime.

[1] David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *ICML*, 2013.
[2] Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*, 2009.
[3] Andrew G. Wilson and Ryan P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *ICML*, 2013.

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2020