Statistical Foundations of Prior-Data Fitted Networks[5]

Author: Prof.Dr. Thomas Nagler

Speaker: Felix Barrez TAMBE NDONFACK

University of Freiburg
Department of Mathematical Stochastics, Mathematical Institute

July 16, 2025

Outline

universität freiburg

Introduction

Literature Review and Motivation for PFNs

Prior-Data Fitted Networks (PFNs), introduced by *Müller et al. (2021)* [3], are a novel machine learning method for tabular data inspired by Bayesian nonparametrics and meta-learning. Built on Transformer architecture, PFNs use set-valued inputs of training and test samples to make predictions in a single forward pass.

PFNs approximate Bayesian inference via in-context learning. Bayesian inference involves updating beliefs about model parameters based on data by computing the posterior $p(\Theta|\mathcal{D})$ and deriving predictions via:

- Via Variational Inference or MCMC, the estimation of $p(\Theta|\mathcal{D}_{\text{train}})$ has been done.
- To make prediction for a new data point x_{test}, [3] compute the predictive distribution as

$$p(y_{\text{test}} \mid x_{\text{test}}, \mathcal{D}_{\text{train}}) = \int_{\Theta} p(\Theta \mid \mathcal{D}_{\text{train}}) p(y_{\text{test}} \mid x_{\text{test}}, \Theta) d\Theta$$

- Instead, PFNs directly approximate $p(y_{\text{test}} \mid x_{\text{test}}, \mathcal{D}_{\text{train}})$ using the following steps:
  1. Sample datasets $\mathcal{D}^{(i)} \sim p(\mathcal{D})$ and split into $\mathcal{D}^{(i)}_{\text{train}}$ and $\mathcal{D}^{(i)}_{\text{test}} = \{x^{(i)}_{\text{test}}, y^{(i)}_{\text{test}}\}$.
  2. Train a model to predict $y^{(i)}_{\text{test}}$ from $\{x^{(i)}_{\text{test}}, \mathcal{D}_{\text{train}}\}$.
  3. The model learns to approximate $p(y_{\text{test}} \mid x_{\text{test}}, \mathcal{D})$ in a single forward pass.

universitätfreiburg

Details for PFNs

Let us have a look at this summary of the illustration of [3]

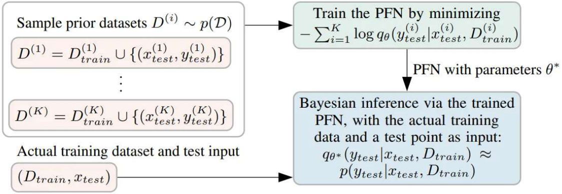

Figure: Visualisation of PFN. We sample datasets from an a priori and fit an PFN to examples of these datasets. Given a real dataset, we feed it and a test point into the PFN and approximate Bayesian inference in a single forward propagation.

Objectives

1. Establish a **theoretical foundation** for Prior-Data Fitted Networks (PFNs):
   - Formalize PFNs as approximations of Bayesian posterior predictive distributions (PPDs).
   - Characterize KL-optimality and Monte-Carlo training dynamics.
2. Identify statistical mechanisms driving **in-context learning**:
   - Decompose error into bias and variance components.
   - Derive conditions for vanishing variance (sensitivity analysis) and bias (locality).
3. Analyze the role of **transformer architecture**:
   - Prove variance decay rates ($O(1/n)$) and bias limits.
   - Validate empirically with TabPFN and localization techniques.

universität freiburg

Research Questions

- **Theoretical Learnability**: Under what prior conditions does the PPD converge to the true $p_0(y \mid x)$?
- **Approximation Quality**: How do the prior Π, size distribution Π_N, and model class $\{q_\theta\}$ influence PFN training?
- **In-Context Learning**: Why can a pre-trained PFN improve predictions on $n > N_{\text{pre-train}}$?
  - Variance: Does transformer symmetry/sensitivity ensure $O(1/n)$ decay?
  - Bias: Why does localization become necessary?
- **Architectural Impact**: How do attention heads in transformers limit bias reduction?

Problem Setup

Formal Setup

Classification Task:

- Features: $X \in \mathcal{X} \subseteq \mathbb{R}^d$, Labels: $Y \in \mathcal{Y}$.
- Observed data: $\mathcal{D}_n = \{(Y_i, X_i)\}_{i=1}^n \sim p_0$, i.i.d.
- Goal: Estimate $p_0(y \mid x) = \mathbb{P}(Y = y \mid X = x)$.

Formal Setup

Classification Task:

- Features: $X \in \mathcal{X} \subseteq \mathbb{R}^d$, Labels: $Y \in \mathcal{Y}$.
- Observed data: $\mathcal{D}_n = \{(Y_i, X_i)\}_{i=1}^n \sim p_0$, i.i.d.
- Goal: Estimate $p_0(y \mid x) = \mathbb{P}(Y = y \mid X = x)$.

Bayesian Nonparametric Framework:

- Treat p_0 as a realization of a random $p \sim \Pi$, where Π is a *prior* over models.
- Posterior predictive distribution (PPD):

$$\pi(y \mid x, \mathcal{D}_n) = \int p(y \mid x)\, d\Pi(p \mid \mathcal{D}_n).$$

- Key assumption: Prior factorizes as $\Pi(p) = \Pi(p(y \mid x)) \cdot \Pi(p(x))$.

universitätfreiburg

Formal Setup

Classification Task:

- Features: $X \in \mathcal{X} \subseteq \mathbb{R}^d$, Labels: $Y \in \mathcal{Y}$.
- Observed data: $\mathcal{D}_n = \{(Y_i, X_i)\}_{i=1}^n \sim p_0$, i.i.d.
- Goal: Estimate $p_0(y \mid x) = \mathbb{P}(Y = y \mid X = x)$.

Bayesian Nonparametric Framework:

- Treat p_0 as a realization of a random $p \sim \Pi$, where Π is a *prior* over models.
- Posterior predictive distribution (PPD):

$$\pi(y \mid x, \mathcal{D}_n) = \int p(y \mid x)\, d\Pi(p \mid \mathcal{D}_n).$$

- Key assumption: Prior factorizes as $\Pi(p) = \Pi(p(y \mid x)) \cdot \Pi(p(x))$.

Prior-Data Fitted Networks (PFNs):

- Goal: Approximate $\pi(y \mid x, \mathcal{D}_n)$ with a parametric model q_θ .
- Training objective (KL-optimality):

$$\theta^* = \arg\max_\theta \mathbb{E}_{\Pi_N} \mathbb{E}_\Pi \left[\log q_\theta(Y \mid X, \mathcal{D}_N)\right],$$

  where $\mathcal{D}_N \sim \Pi$, $N \sim \Pi_N$.
- Architecture: Transformer networks (permutation-equivariant, handles variable n).

PFN Objective and Consistency of PPDs

universität freiburg

PFN Objective: Approximating the PPD

Goal: Learn parametric $q_\theta(y \mid x, \mathcal{D}_n)$ to approximate the PPD $\pi(y \mid x, \mathcal{D}_n)$.

Theorem: Optimality of PPD

$$\pi = \arg\max_{q \in \mathcal{Q}} \mathbb{E}_\Pi \left[\log q(Y \mid X, \mathcal{D}_n)\right],$$

where $\mathcal{Q} = \left\{ q : (\mathcal{Y} \times \mathcal{X})^{n+1} \to [0,1] \,\middle|\, \sum_{y \in \mathcal{Y}} q(y \mid \cdot, \cdot) = 1 \right\}$.

Proof Sketch:

- By definition, π minimizes $\mathrm{KL}(\pi \| q)$.
- Law of iterated expectations and non-negativity of KL divergence.

Training PFNs:

- Monte Carlo approximation of θ^* :

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{m} \log q_\theta(Y_j \mid X_j, \mathcal{D}^{(j)}),$$

where $\mathcal{D}^{(j)} \sim \Pi$, $N_j \sim \Pi_N$.

- Π_N: Size prior (e.g., uniform on $\{1, \ldots, 1023\}$).

Consistency of PPDs

Question: When does $\pi(y \mid x, \mathcal{D}_n)$ converge to $p_0(y \mid x)$?
Assumptions:

- **(A1)**: Prior Π contains a KL-optimal p^*, i.e.,

$$p^* = \arg\min_{p \in \mathcal{P}} \mathrm{KL}(p^* \| p_0).$$

- **(A2)**: Prior mass concentrates near p^* (metric entropy condition).

Theorem: Consistency

$$\pi(y \mid x, \mathcal{D}_n) \xrightarrow{n \to \infty} p^*(y \mid x) \quad \text{a.s. for } p_0\text{-a.e. } (y, x).$$

Proof Sketch:

- Posterior concentration: Show $\Pi(p \mid \mathcal{D}_n)$ concentrates around p^*
- Uses Hellinger metric for convergence.
- Borel-Cantelli lemma : Establish almost sure convergence.

Progress in learning architectures and empirical analyses

universität freiburg

In-Context Learning (ICL)

Definition

Ability of a pre-trained model to adapt to new tasks using only the **context** (input data) provided during inference, *without* parameter updates.

Mechanism in PFNs

- Transformer processes entire dataset $\mathcal{D}_n$ as a sequence.
- Attention heads "attend" to relevant samples in $\mathcal{D}_n$ to predict $P(Y \mid x)$.
- **Key Property**: Predictions improve with larger n (unseen during pre-training).

Why It Works (Theoretically)

- **Variance Decay**: Transformer sensitivity to individual samples diminishes as $n \to \infty$.
- **Structural Bias**: Model architecture implicitly averages over tasks seen during meta-training.

Bias and Locality

Error Decomposition

$$q_\theta(y \mid x, \mathcal{D}_n) - p_0(y \mid x) = \underbrace{\text{Variance}}_{O(n^{1/2-\alpha})} + \underbrace{\text{Bias}}_{\text{Depends on locality}} .$$

Theorem: Locality Necessity

: For bias $\mathbb{E}[q_\theta] - p_0 \to 0$, q_θ must asymptotically depend *only* on samples (X_i, Y_i) near x.

Transformer Limitation:

- Attention weights $a_j^{(h)}$ assign non-zero mass to *all* samples.
- Bias persists unless explicitly localized.

universitätfreiburg

Transformer PFNs: Architecture & Theory

Architecture:

- Input: $\mathcal{D}_n = \{(X_i, Y_i)\}$ + test feature x .
- Multi-head attention: $a_j^{(h)} = \text{SoftMax}(v^\top W_q^{(h)} V_j)$.
- Output: $q_\theta(y \mid x, \mathcal{D}_n) = \text{SoftMax}(W_o z)$.

Theoretical Guarantees:

- **Theorem 6.2:Variance**: Var $\sim O(1/n)$ due to bounded sensitivity.

Theorem 6.3 (Bias Limit)

$$\mathbb{E}[q_\theta] \to \bar{q}_\theta(y \mid x) = \text{SoftMax}\left(W_o \sum_{h=1}^{H} W_v^{(h)} \mathbb{E}_{g_h}[V]\right),$$

where g_h is an exponentially tilted measure.

universität freiburg

Examples of PFN Priors

The performance of PFNs hinges on the design of the synthetic priors $p(D)$ used for pre-training. Notable examples include:

1. **Bayesian Neural Network Prior**: weights are sampled from $\mathcal{N}(0, \sigma^2)$, and datasets are generated by applying the sampled network to random inputs [3]. This prior captures uncertainty over neural network parameters.
2. **Gaussian Process Prior**: Hyperparameters (e.g. length scales) are sampled from a meta-prior, and datasets are drawn from the resulting GP. Enables PFNs to approximate GP to approximate GP inference while avoiding cubic complexity [3].
3. **TabPFN Prior**: A structural causal model prior with linear relationships and node-specific noise, tailored for tabular data. Achieves state-of-the-art performance on small tabular datasets [2]. It includes Time-series Prior to incorporate seasonality and trends to specialize PFNs for forecasting [1].

These priors demonstrate how domain knowledge can be encoded into PFNs through synthetic data generation, enabling applications ranging from genomics to Bayesian optimization (For further reading, see [4]).

universität freiburg

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White.
Forecastpfn: Synthetically-trained zero-shot forecasting.
Advances in Neural Information Processing Systems, 36:2403–2426, 2023.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter.
Tabpfn: A transformer that solves small tabular classification problems in a second.
arXiv preprint arXiv:2207.01848, 2022.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter.
Transformers can do bayesian inference.
arXiv preprint arXiv:2112.10510, 2021.

Samuel Müller, Arik Reuter, Noah Hollmann, David Rügamer, and Frank Hutter.
Position: The future of bayesian prediction is prior-fitted.
arXiv preprint arXiv:2505.23947, 2025.

Thomas Nagler.
Statistical foundations of prior-data fitted networks.
In *International Conference on Machine Learning*, pages 25660–25676. PMLR, 2023.