Review on TabPFN

TabPFN: A transformer that solves small tabular classification problems in a second [1]

presentor: Jihu Lee

IDEA lab
Department of Statistics
Seoul National University

March 20, 2025

What is TabPFN?

- A foundation model for tabular dataset.
- PFN (Prior-Data Fitted Networks)[2] on tabular dataset.
- A single transformer - supervised classification for small tabular datasets in *less than a second*.

What is PFN?

- Paper: Transformers Can Do Bayesian Inference [2]
- Goal: approximate a large set of posteriors (to be explained. . .)

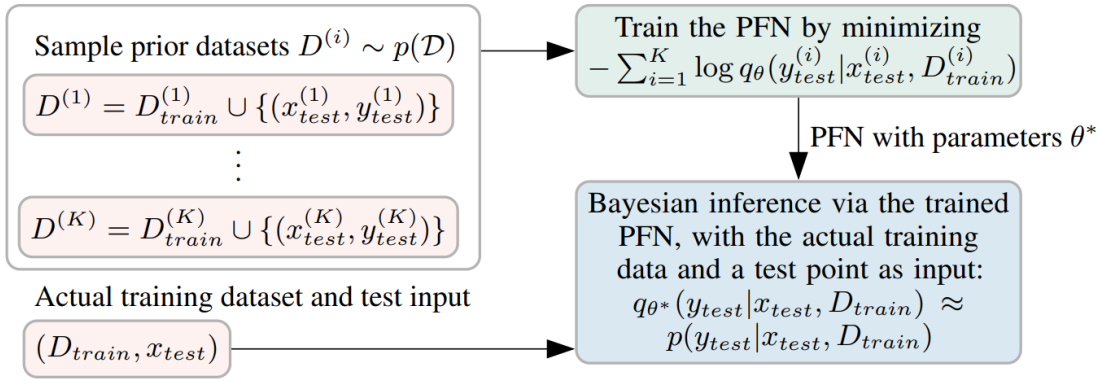


Figure 1: A visualization of Prior-Data Fitted Networks (PFNs). We sample datasets from a prior and fit a PFN on hold-out examples of these datasets. Given an actual dataset, we feed it and a test point to the PFN and obtain an approximation to Bayesian inference in a single forward propagation.

What is PFN?

- Train set: $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$
- New data and its output: $(x_{\text{test}}, y_{\text{test}})$.
- Goal: approximate Posterior predictive distribution (PPD)

$$p(y_{\text{test}}|x_{\text{test}}, D_{\text{train}}) \tag{1}$$

- How to approximate? **Prior-Data fitting**

Prior-Data Fitting

- Loss: *Prior-Data NLL*

$$
\begin{aligned}
L_{PFN}(\theta) &= \mathop{\mathbb{E}}_{(\{(x_{\text{test}}, y_{\text{test}})\} \cup D_{\text{train}}) \sim p} [-\log q_\theta(y_{\text{test}} | x_{\text{test}}, D_{\text{train}})] \\
&= \mathbb{E}_{x, D_{\text{train}} \sim p} [H(p(\cdot | x, D_{\text{train}}), q_\theta(\cdot | x, D_{\text{train}}))]
\end{aligned} \tag{2}
$$

- H: cross-entropy
- Practically, authors choose q_θ as a transformer.

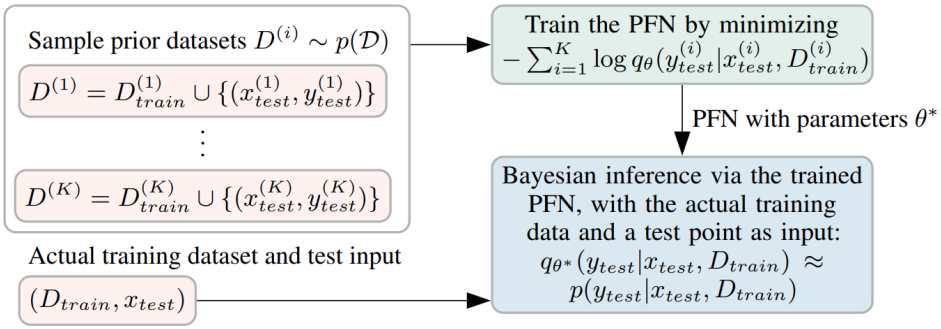


Figure 1: A visualization of Prior-Data Fitted Networks (PFNs). We sample datasets from a prior and fit a PFN on hold-out examples of these datasets. Given an actual dataset, we feed it and a test point to the PFN and obtain an approximation to Bayesian inference in a single forward propagation.

Variational Inference vs. PFN

VI

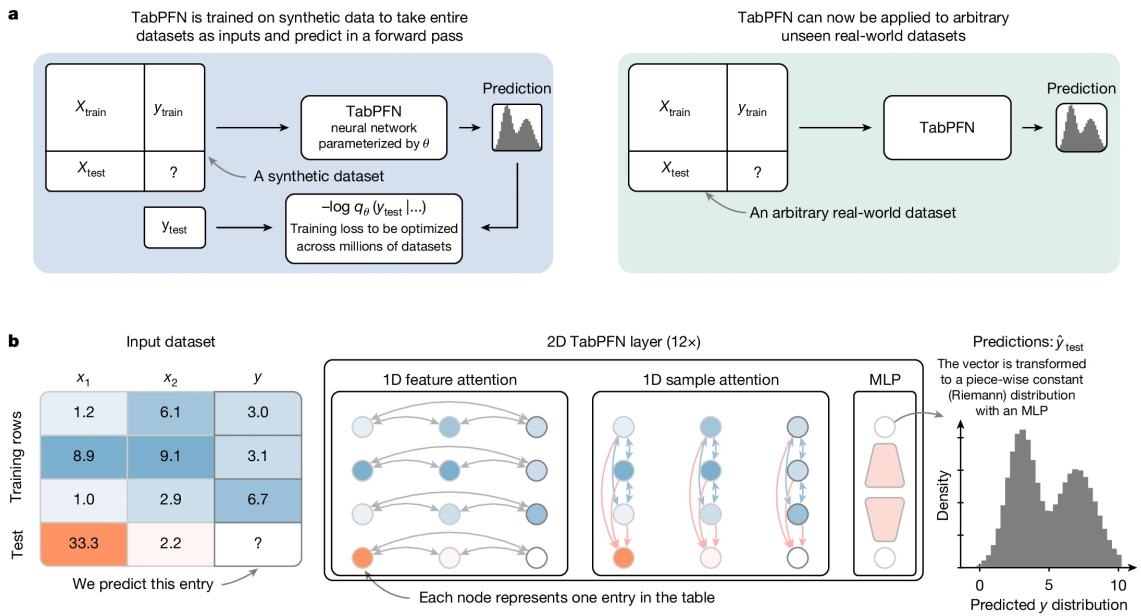
- $q_\phi(w) \approx p(w|D)$ (w: weight parameter)
- $p(y^*|x^*, D) \approx \int p(y^*|x^*, w) q_\phi(w) dw$
- When D differs, ϕ should be re-trained.

PFN

- Directly approximate $q_\theta(y|x, D) \approx p(y|x, D)$, in $(D, x) \mapsto y$ way.
- $p(y^*|x^*, D) \approx q_\theta(y^*|x^*, D)$
- Does not require a re-train $\rightarrow$ fast.

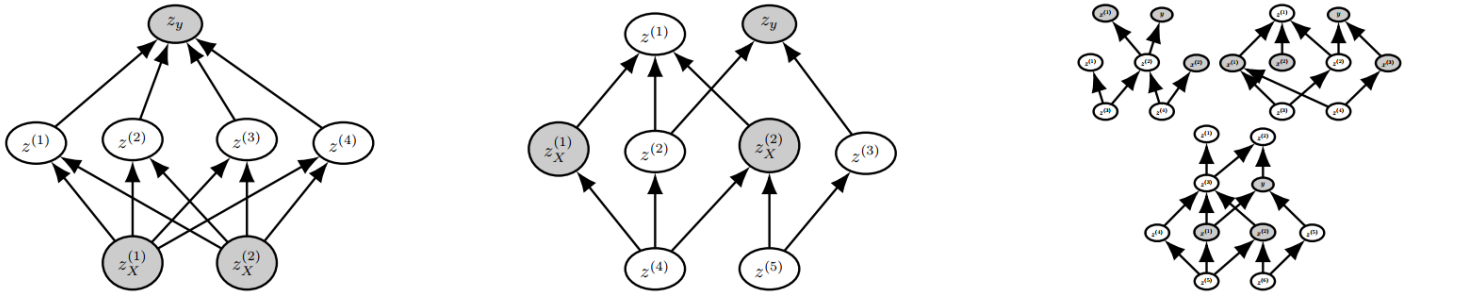
TabPFN vs. PFN

- TabPFN is a PFN, but with a **novel prior for tabular data**.
- Technical Modification:
  i) Modify attention masks.
  ii) Enable the model to work on datasets with different number of features, by zero-padding.



A Prior for Tabular Data

BNN and SCM prior



(a) A BNN (b) An SCM (c) SCMs sampled from the prior

Figure 2: Overview of graphs generating data in our prior. Inputs x are mapped to the output y through unobserved nodes z. Plots based on Müller et al. (2022).

- SCM = Structured Causal Model.

BNN Prior

- Prior on architecture
  (1) sample a model architecture: $A \sim p(A)$
  (2) sample model weights, given A: $W_{i,j} \sim p_w(\cdot)$
  (3) sample i.i.d. features $x_{i,f} \sim N(0,1)$ (i: index, f: feature)
  (4) yield $\{(x_i, A_W(x_i))\}_{i=1}^N$

SCM Prior

- Tabular data: causal relationships between columns.
- Sample DAG structure & deterministic functions.
  (1) Sample MLP structure and its weights.
  (2) Drop a random set of edges (MLP with dropped edges = DAG structure).
  (3) Sample feature nodes and a label node.
  (4) Sample noise distribution $p(\epsilon) \sim p(p(\epsilon))$.
  (5) Sample noise variables ϵ_i.
  (6) Compute node values as $z_i = a((\sum_{j \in PA_{\mathcal{G}(i)}} E_{ij} z_j) + \epsilon_i)$.
  (7) Retrieve the values of feature nodes and the output node.

Toy Data

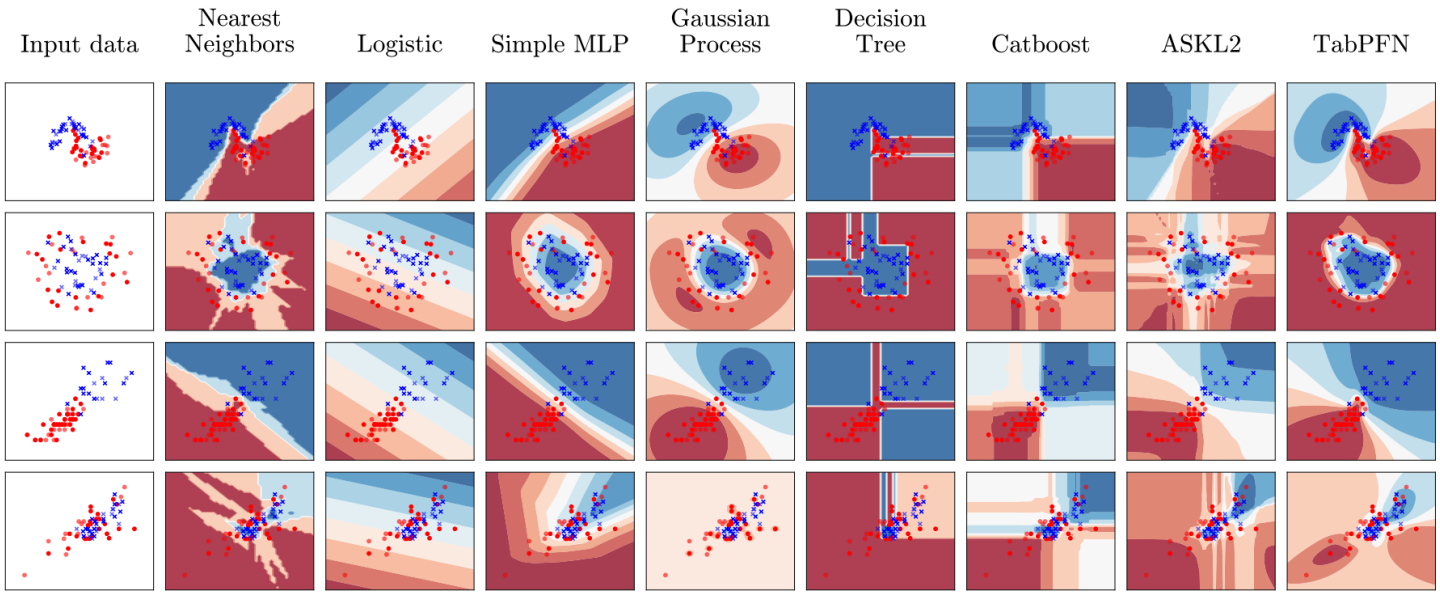
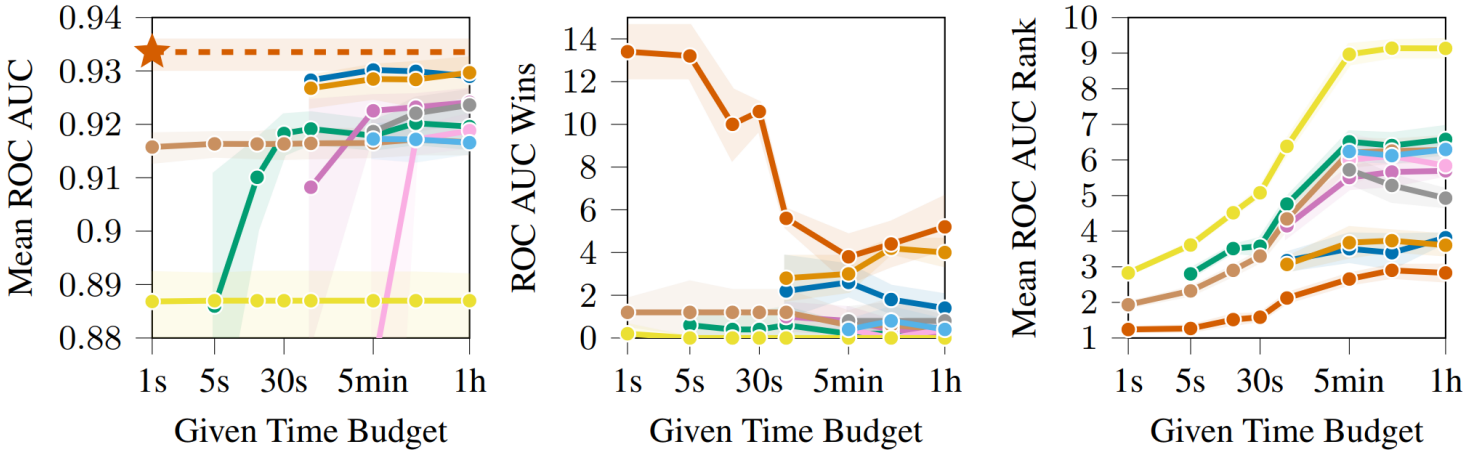


Figure 4: Decision boundaries on toy datasets generated with *scikit-learn* (Pedregosa et al., 2011).

Real Datasets



Pros & Cons

Pros

i) Fast inference.

ii) No need of additional training.

iii) Bayesian inference.

iv) Understanding on the causal structure of data generation.

Cons

i) Cannot applied in large datasets.

ii) Hard to handle categorical variables.

iii) Hard to handle high-dimensional (>100) datasets.

References

[1] Hollmann, Noah, et al. "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". International Conference on Learning Representations (2023).

[2] Müller, Samuel, et al. "Transformers Can Do Bayesian Inference." International Conference on Learning Representations (2022).


THANK
You!