예시 Beamer 타입의 LaTeX문서

Sungwoo Park

Uncertainty Quantification Lab
Seoul National University

August 19, 2025

Table of contents

Contents

Beamer 작성 시 유의할 점

- Beamer는 프레젠테이션을 위한 LaTeX 클래스입니다.
- SNU Beamer 테마를 사용한 템플릿입니다.
- 도큐먼트와 다르게, 해당 파일도 같이 있어야 합니다.
  - beamerthemesnubeam.sty
  - snucode.sty
  - ref.bib
- 각 파일의 쓸모는 알아서 잘 찾아보세요...

Brief Introductions about TabPFN

- TabPFN: A Transformer that solves small tabular classification problems in a second [3]
- TabPFNv2: Accurate predictions on small data with a tabular foundation model [4]
- Fast tabular statistical model using transformer.
- The distribution of the prediction is calculated in this formula.

$$p(y|x, D) = \int_{\Phi} p(y|x, \phi)p(D|\phi)p(\phi)d\phi$$

Structure of TabPFN

1. Pre-training
   - Model is trained on synthetic datasets (created by SCM) by minimizing the discrepancy between the predicted label of the test instance and its true label
   - We don't know the pre-training phase's structure : In the pip library, it just downloads pre-trained parameters and performs the inference phase.
2. Inference
   - For given data (training set + test set), TabPFN performs a single forward pass to generate predictions for all test samples.
   - In inference phase, model only do in-context learning, not backpropagation

TabPFN Example

Here is an example for fitting and making a prediction using TabPFN:

```
> from tabpfn import TabPFNRegressor
> ....
> train_1 = pd.DataFrame(data = [[-2, -2], [-1, -1], [1, 1], [2, 2]],
lumns = ['x', 'y'])
> test = pd.DataFrame(data = [[0]], columns = ['x'])
> regressor = TabPFNRegressor(device = 'cuda')
> regressor.fit(train_1[['x']], train_1[['y']]) // Forward pass
> regressor.predict(test) // Prediction
  array([0.1495245], dtype=float32)
```

$$\begin{array}{|c|}\hline \mathbf{x}\\\hline -2\\\hline -1\\\hline 1\\\hline 2\\\hline\end{array}, \begin{array}{|c|}\hline \mathbf{y}\\\hline -2\\\hline -1\\\hline 1\\\hline 2\\\hline\end{array} \xrightarrow{\text{TabPFNRegressor}} \text{regressor}, \begin{array}{|c|}\hline \mathbf{x}\\\hline 0\\\hline\end{array} \xrightarrow{\text{regressor}} \begin{array}{|c|}\hline \mathbf{y}\\\hline 2\\\hline\end{array}$$

Contents

Is TabPFN better then traditional statistical models?

- In the past, many studies show that traditional statistical models (XGBoost, Random Forest, etc.) still outperform most deep learning models on tabular data. [6], [2]
- But TabPFN claims that their model is a state-of-the-art model for tabular data.

Why does this happen?

- TabPFN excels in handling small- to medium-sized datasets with up to 10,000 samples and 500 features. For larger datasets and highly non-smooth regression datasets, approaches such as CatBoost, XGB or AutoGluon are likely to outperform TabPFN. [4]
- The model was trained on a synthetic dataset generated via an SCM with about 130 million dataset samples.
- Since the dataset covered only a limited value range, the model did not learn to generalize beyond that domain.

TabPFN-TS

- Hoo et al., 2025 proposed TabPFN-TS, a TabPFN variant specific for time series.
- Proposes optimal transformation (preprocess) strategies tailored for TabPFN with time series data.

TabPFN-TS: Preprocessing

- Running index: chronological order
- Calendar features: second, minute, hour, weekday, day (month), day (year), week (year), year
- Seasonal features:
  - Additional cycles (e.g., lunar, non-calendar periodicities)
  - FFT top-k frequency features ($k = 5$)
- Cyclic features: sin / cos transformation

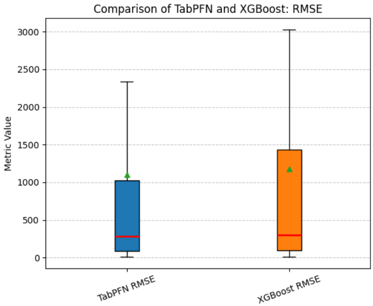
$$\Phi(t) = \left(\cdots, \cos\left(\frac{2\pi t}{P_i}\right), \sin\left(\frac{2\pi t}{P_i}\right), \cdots \right)$$

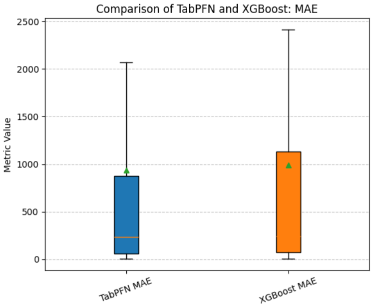
TabPFN-TS vs XGBoost

- In the paper, there is no comparison with XGBoost
- We conducted experiments using the Chronos [1] m4_hourly 100 time series datasets, predicting the last 48 data points.
- For a fair comparison, XGBoost was trained with the same feature set as TabPFN-TS.

TabPFN-TS vs XGBoost

- hi

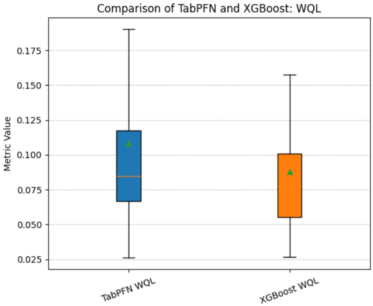




- hello

TabPFNv2 vs others

- Hello



Contents

Conclusion

- 안녕

Thank you!

References I

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang.
Chronos: Learning the language of time series.
arXiv preprint arXiv:2403.07815, 2024.

[2] Léo Grinsztajn, Edouard Oyallon, and G Varoquaux.
Why do tree-based models still outperform deep learning on typical tabular data?
35:507–520.

References II

[3] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter.
TabPFN: A transformer that solves small tabular classification problems in a second.

[4] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter.
Accurate predictions on small data with a tabular foundation model.
637:319–326.

[5] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter.
From tables to time: How TabPFN-v2 outperforms specialized time series forecasting models.

References III

[6] Ravid Shwartz-Ziv and Amitai Armon.
Tabular data: Deep learning is not all you need.

[7] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao.
A closer look at TabPFN v2: Understanding its strengths and extending its capabilities.