Scaling TabPFN: Sketching and Feature Selection for Tabular Prior-Data Fitted Networks

Benjamin Feuer, Chinmay Hegde, Niv Cohen
New York University
{bf996, chinmay.h, niv.cohen}@nyu.edu

Abstract

Tabular classification has traditionally relied on supervised algorithms, which estimate the parameters of a prediction model using its training data. Recently, Prior-Data Fitted Networks (PFNs) such as TabPFN have successfully learned to classify tabular data *in-context*: the model parameters are designed to classify new samples based on labelled training samples given *after* the model training. While such models show great promise, their applicability to real-world data remains limited due to the computational scale needed. Here we study the following question: given a pre-trained PFN for tabular data, what is the best way to summarize the labelled training samples before feeding them to the model? We conduct an initial investigation of sketching and feature-selection methods for TabPFN, and note certain key differences between it and conventionally fitted tabular models.

1 Introduction

Classification of tabular data is a basic machine learning task of vital importance, and has inspired a large set of algorithmic approaches. These approaches, ranging from classical algorithms such as gradient-boosted trees [Prokhorenkova et al., 2018; Chen & Guestrin, 2016] to recent attempts with deep learning [Somepalli et al., 2021] have all been focused on choosing proper model parameters (and hyper-parameters) given an explicitly defined model hypothesis class.

In a parallel line of work, *Prior-Data Fitted Networks* such as TabPFN have successfully been demonstrated to classify tabular data based on a training set given as the model *input* [Hollmann et al., 2023]. Rather than using training data to fit the model parameters, TabPFN gets at inference time a set that contains *both* labeled and unlabeled samples. It then predicts the 'missing' labels directly based on the labeled input samples, rather than exclusively relying on trained model parameters. In this sense, the working of TabPFN resembles the phenomenon of *in-context learning* that is exhibited by large language models such as GPT, which emerges during pretraining [Brown et al., 2020; Li et al., 2023]. In TabPFN, the pretraining stage can be viewed as inducing suitable statistical priors which are then used to fill in the missing labels.

In large language models, the choice (and length) of the context crucially affects prediction performance [Xie et al., 2022]. Optimizing the performance of large language models often requires long (and fairly complicated) prompting strategies. Many recent studies have examined how prompting in-context learning algorithms allows the model to retrieve memorized information, perform simple mathematical operations, or even execute high-level reasoning [Liu & Low, 2023; Meng et al., 2022; Nanda et al., 2023]. Context-optimization strategies have been less frequently studied for tabular models such as TabPFN. They are, however, crucial, as memory constraints prevent TabPFN from using more than a few thousand samples as context.

When using Prior-Data Fitted Networks for tabular data, the "prompt" is composed from the sample values and their labels from the training set. The model is then expected to give predictions on a

2nd Table Representation Learning Workshop: 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

set of unlabelled samples, also given as part of the prompt. This setting is simpler than the case for language models: all the labelled parts of the prompt are weighed equally, and the labeled (training) samples can be treated as an unordered set, rather than parts of an ordered string with internal syntax.

Focusing on in-context learning for tabular data has additional advantages: (i) While the evaluation of language models is a hard problem (covering many tasks and complicated metrics) the evaluation of tabular data classifiers is straightforward. (ii) the prior knowledge implicitly contained in the pretrained model is driven in the tabular case by synthetic data; unlike the case with large language models which use large text corpora, which are not as tractable. (iii) Improving the results of TabPFN supports the study of deep models for tabular data and may be of practical importance.

Here, we focus on understanding one part of the context optimization problem for tabular classification: learning to summarize a large training dataset, X_{labelled}, to a more compact set, X_{compact}, such that most of the information useful to the model is contained in X_{compact}. The relative simplicity of tabular in-context learning allows us to formalize a concrete question: Given a pretrained model, which function should we use to scale input target data before feeding it to the model?

In this short paper, we empirically study basic properties about how one should (or should not) summarize a target tabular dataset when used as context for in-context learning.

2 Sketching and feature selection for tabular in-context learning

Feature selection and sketching methods have been extensively explored in prior literature on tabular classification [Munteanu & Schwiegelshohn, 2018; Chandrashekar & Sahin, 2014]. In the following section, we report the results of applying a representative selection of these methods.

2.1 Experimental setting

To systematically evaluate TabPFN with the different context summarization we select a subset of nineteen datasets from [McElfresh et al., 2023] which exceed either the feature or sample limitation of TabPFN, which are 100 features and 1000 samples, respectively [Hollmann et al., 2023]. We limit our algorithmic comparison to TabPFN and CatBoost, which is the best-performing overall model in [McElfresh et al., 2023]. The complete list of datasets can be found in Sec. 6. For ease of comparison with existing meta-analyses, where possible we replicate the method of [McElfresh et al., 2023]. We compare CatBoost with 30 hyperparameter settings (one default set and 29 random sets, using Optuna) to TabPFN, averaging over 10 train/validation folds for each dataset.

Our main results can be found in Tab. 1. We conduct our primary investigation into sketching methods at $d_{\text{max}} = 100$ features and $n_{\text{max}} = 3000$ samples from each dataset. Dataset names are drawn from OpenML with abbreviations as follows: gddc refers to gas-drift-different-concentrations, fm to Fashion-MNIST, ss to skin-segmentation. CB stands for CatBoost, TP for TabPFN, f for full dataset, b for best result using any combination of algorithms, and r for the best result using random feature and random sample selection. SKT is an abbreviation for sketching method, FTS for feature selection, SMP for sampling strategy. We report the most successful combination for each model, on each dataset, with respect to average accuracy over ten folds.

We determine statistically significant ($p < 0.05$) performance differences between algorithms by use of a Wilcoxon signed-rank test between CatBoost and TabPFN. A Holm-Bonferroni correction is used to account for multiple comparisons.

2.2 The effect of scale for tabular data classification

We begin with a short empirical study of the effect of the number of supplied labelled samples on the accuracy of our compared algorithms (TabPFN and CatBoost), noting the following interesting facts: (i) The authors of TabPFN do not recommend using the model beyond its 1000-sample context length limit [Hollmann et al., 2023] without retraining. In Fig. 1, we ablate the effect of using different quantities of samples. We consider 100, 500, and 3000 samples for both CatBoost and TabPFN. Whiskers represent one standard deviation from the mean. Feature subsets are taken using mutual information, sampling is random, and class weighting is proportional.

We find that performance improves on most datasets when increasing the context length to 3000 samples using TabPFN, but the gains are far more pronounced in CatBoost. (ii) We find that TabPFN often outperforms CatBoost at sample sizes up to 1000, and remains highly competitive above that

Dataset	Acc (CB, f)	Acc (CB, b)	Acc (CB, r)	Acc (TP, b)	Acc (TP, r)	SKT / FTS / SMP (CB)	SKT / FTS / SMP (TP)
airlines_189354	**0.653**	0.637	0.637	0.594	0.589	RND / RND / PR	RND / RND / PR
albert_189356	**0.698**	0.657	0.657	0.64	0.64	RND / RND / PR	RND / RND / PR
CIFAR_10_167124	**0.434**	0.37	0.342	0.373	0.372	RND / PCA / PR	RND / RND / PR
connect-4_146195	**0.749**	0.716	0.716	0.66	0.659	RND / RND / PR	RND / RND / PR
eeg-eye-state_14951	0.832	0.808	0.806	**0.932**	**0.932**	RND / RND / PR	RND / RND / EQ
elevators_3711	0.855	0.845	0.838	**0.9**	**0.899**	CLS / MUT / PR	RND / RND / PR
Fashion-MNIST_146825	**0.843**	0.787	0.787	**0.835**	0.812	RND / RND / PR	RND / PCA / PR
gas-drift-different-concentrations_9987	0.97	0.976	0.955	**0.994**	**0.993**	RND / PCA / EQ	RND / RND / PR
higgs_146606	**0.71**	0.684	0.684	0.665	0.661	RND / RND / PR	RND / RND / PR
hill-valley_145847	0.514	0.514	0.514	**0.56**	**0.56**	RND / RND / PR	RND / RND / PR
mfeat-factors_12	0.954	0.95	0.943	**0.973**	**0.973**	KMN / RND / EQ	RND / RND / PR
mfeat-pixel_146824	0.955	0.951	0.951	**0.971**	**0.97**	RND / RND / PR	RND / RND / PR
pendigits_32	0.972	0.966	0.964	**0.995**	**0.993**	RND / RND / PR	RND / RND / PR
poker-hand_9890	**0.664**	0.572	0.561	0.519	0.515	RND / RND / PR	RND / RND / PR
riccardo_168338	0.951	0.956	0.93	**0.991**	0.982	RND / PCA / EQ	RND / MUT / EQ
robert_168332	**0.446**	0.367	0.367	0.384	0.359	RND / RND / PR	RND / PCA / EQ
semeion_9964	0.887	0.869	0.863	**0.915**	**0.915**	RND / MUT / EQ	RND / RND / PR
skin-segmentation_9965	**0.994**	0.989	0.987	**0.999**	**0.999**	RND / RND / PR	RND / RND / PR
volkert_168331	**0.608**	0.56	0.56	0.557	0.555	RND / RND / PR	RND / RND / PR

Table 1: ***Comparative performance of TabPFN (TP) and CatBoost (CB) with sketching methods.*** *We compare at a fixed feature size of 100 and a fixed sample size of 3000. When both models are limited to 3000 samples, TabPFN performs better on 12 of 17 datasets where significant differences exist. When Catboost is allowed access to the entire training data, the win rate is identical. In most cases, random sample selection is sufficient for optimal performance. Both models benefit from PCA and mutual information dimensionality reduction when the feature space is large.* ***Bold*** *indicates the best-performing model(s).*

threshold. That said, we acknowledge that with a more rigorous parameter search, either model may be capable of exceeding our reported metrics (achieved using random hyperparameter sweeps.)

2.3 Sketching for tabular in-context learning

The context length often limits the usefulness of TabPFN (Fig.1). Therefore, when reducing the given number of samples to a smaller subset, we wish the subset to preserve as much useful context from the original dataset. Since the utilization of context by the model T is not explicitly known, we turn to empirical investigation. The summarization of context samples may depend on the samples themselves X_{labelled}, their labels y, or both. We study these factors independently: we examine summarization methods for X_{labelled}, and apply them according to the labels y in one of two manners: *equal*: having a similar amount of samples from each y label; and *proportion*: keeping the number of samples from each label proportional to their abundance in the original data.

In terms of the samples summarization method, we investigate a few options: *random:* picking a random subset of samples. *K-means:* Choosing the samples as the K-means cluster centers of the original data, where K is determined according to $n_{\max}$. *CoreSet Agarwal et al. [2005]:* Choosing an $n_{\max}$ sized CoreSet. *Closest:* Taking the $n_{\max}$ closest point to a random reference point from the validation set. We implement our methods using the `faiss` library. [Johnson et al., 2019]

Results. We summarize the results in Tab. 1. In most cases, we find that random selection of samples works as well as any other method, making it a strong baseline for future experiments. CatBoost does show a statistically significant benefit from sketching on the *elevators* and *mfeat-factors* datasets. Intriguingly, the latter only has 2000 samples in the dataset; it is probable that sampling with replacement using the equal strategy leads to a better sample emphasis with K-means, compared to random sampling (by having a less biased model). Sampling y values in an equal (vs. proportional) manner is beneficial on 21% of datasets when using CatBoost, and on 16% of datasets when using TabPFN. In some cases, the difference is quite large; the best context-summarization for TabPFN / *riccardo* using equal sampling attained an accuracy of 99%, compared to just 80% when using proportional sampling.

2.4 Feature dimensionality reduction for tabular in-context learning

We investigate three options for feature dimensionality reduction: *random reduction:* picking a random subset of features. *mutual information:* selects features with high mutual information to the target dataset [Vergara & Estévez, 2014]. *PCA:* taking the $d_{\max}$ first principal components. We use the scikit-learn implementations for both methods [Pedregosa et al., 2011].

We find that feature subsampling often has a significant effect on classifier performance, and that the in-context classification method of TabPFN is more dependent on feature selection than that of CatBoost. 4 of our datasets have more than 256 features; (*riccardo*, *Robert*, *Fashion-MNIST*, *CIFAR-*

10). For *Robert*, *FashionMNIST*, and *riccardo*, the best TabPFN setting with mutual information or PCA feature selection outperforms random reduction. See Tab. 1.

On a related note, we report in Fig. 1 the mean normalized performance for both TabPFN and CatBoost with 10, 30, and 100 features. Whiskers represent one standard deviation from the mean. Feature subsets are taken using mutual information, sampling is random, and class weighting is proportional. While both models improve with more samples, the effect is more pronounced and consistent in CatBoost; as feature quantity scales up, performance is the same or better on all datasets. When we reduce the feature space in TabPFN from 100 to 10 features, performance *improves* on two datasets (*higgs* and *connect-4*). This indicates that in-context tabular classification may be more sensitive to the presence of spurious features than supervised methods.

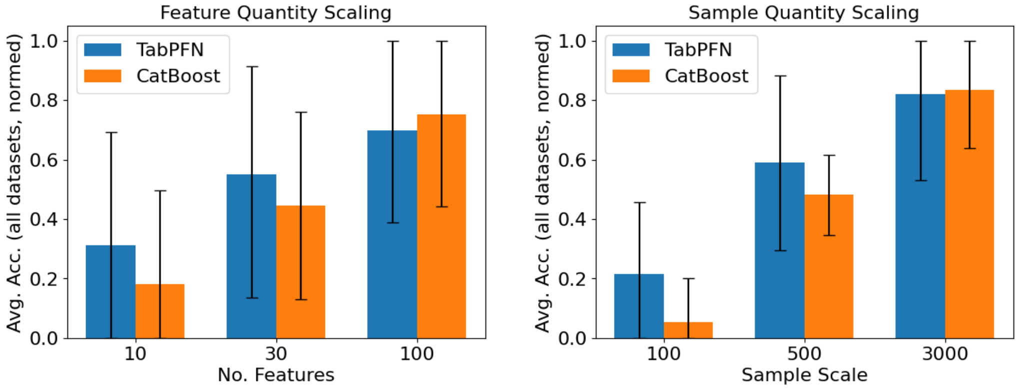

Figure 1: ***(L) Effects of feature quantity scaling. (R) Effects of sample quantity scaling.*** *(L) While both models benefit from feature quantity scaling from 10 to 100 features seen, the gains are more pronounced and consistent in CatBoost. (R) Performance gains in TabPFN are more gradual than in CatBoost as sample quantity increases from 100 to 3000 training samples seen, and less consistent. (Both) The y-axis represents the average normalized accuracy across all datasets. In the case that a given dataset has fewer samples or features than our quantity scaling figure, we take all samples or features from that dataset. The error bars represent one standard deviation from the mean. Image best viewed in color.*

3 Related works

Evaluation of deep learning for tabular data.[Borisov et al., 2022] introduced into the literature TabSurvey, an exhaustive comparison between existing deep learning approaches for heterogeneous tabular data. Their work was extended recently by [McElfresh et al., 2023]. Like our work, the latter directly compares CatBoost and TabPFN with sample and feature sketching. We extend this considerably, demonstrating the effects of data scaling and the comparative importance of feature and sample selection methods.

Learning on a budget. Another relevant line of works considered "learning on a budget" Hacohen et al. [2022]; Yehuda et al. [2022], where one aims to label an optimal subset of a set of samples to train a supervised model effectively.

Efficient attention mechanisms for transformers. The runtime and memory usage of transformer architectures such as TabPFN grows quadratically with the number of training samples, placing severe computational bounds on their ability to scale. A wide range of efficient attention mechanisms have been proposed as a workaround for this limitation. Press et al. [2021]; Chen et al. [2023]; Sun et al. [2022]; Xiao et al. [2023]; Han et al. [2023] We consider such mechanisms an important direction for future research into scaling TabPFN.

Feature selection and data valuation. Other related fields study similar problem to ours for other reasons. Feature selection and feature extraction aim to select a subset of features (or combinations of them) to achieve various goals. Such goals include building simpler, faster, and more comprehensible models [Li et al., 2017; Kumar & Minz, 2014; Zebari et al., 2020]. Our initial study on feature selection or extraction for TabPFN and the examined methods fall well within the feature these fields of study; suggesting new applications to it. We expect the nature of this new application, to highlight new feature selection techniques, which might differ from the optimal techniques used for other goals. The field of data valuation aims to quantify the economic value of different parts of a dataset to fairly compensate different data contributors Jia et al. [2019]; Sim et al. [2022]; Fleckenstein et al. [2023]. Here, our focus is on giving the optimal context for a predictor, rather than understanding the fair value of each individual part.

4 Conclusions

Taken together, our results show that TabPFN can be scaled to perform competitively with CatBoost. First, in terms of the number of used samples, although smart representative selection methods do not boost the results significantly, simply selecting a larger subset than the one used by the authors leads to significant gains. Second, TabPFN can benefit considerably from careful feature selection, more so than CatBoost. We believe that finding further connections between increasing the effective context of tabular prior fitted networks and optimizing the context of language in-context-learning is an exciting future research direction.

TabPFN is capable of performing competitively with CatBoost on larger datasets than its authors indicate. That said, it benefits considerably from careful feature selection, more so than CatBoost. We believe that connections between tabular context and language in-context-learning (and vice versa) are an exciting future research direction.

References

Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1):1–30, 2005.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022. doi: 10.1109/TNNLS.2022.3229161.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, January 2014. ISSN 0045-7906. doi: 10.1016/j.compeleceng.2013.11.024. URL `https://www.sciencedirect.com/science/article/pii/S0045790613003066`.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL `http://doi.acm.org/10.1145/2939672.2939785`.

Mike Fleckenstein, Ali Obaidi, and Nektaria Tryfona. A Review of Data Valuation Approaches and Building and Scoring a Data Valuation Model. *Harvard Data Science Review*, 5(1), jan 26 2023. https://hdsr.mitpress.mit.edu/pub/1qxkrnig.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. *arXiv preprint arXiv:2202.02794*, 2022.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P. Woodruff, and Amir Zandieh. HyperAttention: Long-context Attention in Near-Linear Time, October 2023. arXiv:2310.05869 [cs].

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second, May 2023. URL `http://arxiv.org/abs/2207.01848`. arXiv:2207.01848 [cs, stat].

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Vipin Kumar and Sonajharia Minz. Feature selection: a literature review. *SmartCR*, 4(3):211–229, 2014.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task, 2023.

Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned LLaMA Outperforms GPT-4 on Arithmetic Tasks, May 2023. URL `http://arxiv.org/abs/2305.14201`. arXiv:2305.14201 [cs].

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When Do Neural Nets Outperform Boosted Trees on Tabular Data?, May 2023. URL `http://arxiv.org/abs/2305.02997`. arXiv:2305.02997 [cs, stat].

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

Alexander Munteanu and Chris Schwiegelshohn. Coresets-Methods and History: A Theoreticians Design Pattern for Approximation and Streaming Algorithms. *KI - Künstliche Intelligenz*, 32 (1):37–53, February 2018. ISSN 1610-1987. doi: 10.1007/s13218-017-0519-3. URL `https://doi.org/10.1007/s13218-017-0519-3`.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. arXiv, January 2023. URL `http://arxiv.org/abs/2301.05217`. arXiv:2301.05217 [cs].

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://papers.nips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html`.

Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:“ingredients”, strategies, and open challenges. In *Proc. IJCAI*, pp. 5607–5614, 2022.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. *arXiv preprint arXiv:2212.10554*, 2022.

Jorge R Vergara and Pablo A Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24:175–186, 2014.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks, September 2023. arXiv:2309.17453 [cs].

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference, July 2022. URL `http://arxiv.org/abs/2111.02080`. arXiv:2111.02080 [cs].

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. *Advances in Neural Information Processing Systems*, 35:22354–22367, 2022.

Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2):56–70, 2020.

5 Discussions

Why is feature dimensionality reduction more impactful than sketching? While a complete rigorous investigation of this result is left for future work, we do note that the sample size is typically very large (e.g., 3000) even after sketching, while the feature dimension is often reduced from a few thousands to merely 100 features. We hypothesis that a very large set size allows some sort of statistical convergence in the sample dimension, not occurring when sub-sampling a large feature space to 100 features in a naive way.

6 Complete list of datasets

In Tab. 2, we list some key features of the nineteen datasets included in our meta-analysis. Additional information on the datasets is available from the OpenML website.

dataset	**n. classes**	**n. features**	**n. samples**	**pct. samples seen @ 3000**
airlines_189354	2	7	539383	0.6
albert_189356	2	78	425240	0.7
CIFAR_10_167124	10	3072	60000	5
connect-4_146195	3	42	67557	4.4
eeg-eye-state_14951	2	14	14980	20
elevators_3711	2	18	16599	18.1
Fashion-MNIST_146825	10	784	70000	4.3
gas-drift-different-concentrations_9987	6	129	13910	21.6
higgs_146606	2	28	98050	3.1
hill-valley_145847	2	100	1212	100
mfeat-factors_12	10	216	2000	100
mfeat-pixel_146824	10	240	2000	100
pendigits_32	10	16	10992	27.3
poker-hand_9890	10	10	1025009	0.3
riccardo_168338	2	4296	20000	15
robert_168332	10	7200	10000	30
semeion_9964	10	256	1593	100
skin-segmentation_9965	2	3	245057	1.2
volkert_168331	10	180	58310	5.1

Table 2: **Complete list of datasets.**

7 Formal statement of our optimization problem

For completeness, we provide here a mathematical formulation of our optimization objective.

Given a pretrained model T, which function S should we use to scale input target data before feeding it to the model? Specifically, for an m-class classification problem, we wish to reduce the size of the target labelled dataset $X \in \mathcal{R}^{n \times d}$, and labels $y \in \{0, \ldots, m\}^n$ such that the sample number would be limited by n_{max} and the feature dimension by d_{max}. We describe the reduction of the labelled set as a $(X_{\text{compact}}, y_{\text{compact}}) = S(X_{\text{labelled}}, y)$. We wish to find a function S such that using it the unlabelled sample X_{test} would be correctly classified by the prompted model T, according to their ground truth labels t_{test} (not given during training):

$$\max_{S}\Big(\big(\text{Accuracy}(T(X_{\text{compact}}; y_{\text{compact}}; X_{\text{test}}), y_{\text{test}}\big)\Big)$$

$$\text{s.t. } X_{\text{compact}} \in \mathbb{R}^{n' \times d'}, \quad d' \leq d_{\text{max}}, \quad n' \leq n_{\text{max}}.$$