The Effectiveness of Personalised Federated Learning Methods on Variations of Non-IID Data

Euodia Dodd
Department of Computer Science
University of Cambridge
ed581@cam.ac.uk

Abstract

Despite the recent advances in federated learning, the problem of dealing with heterogeneous data is still a substantial hurdle. In recent years, personalised federated learning methods have been developed to capitalise on the diversity of client data by creating more personalised models for individual clients. However, how these methods cope with varying types of heterogeneity has yet to be investigated. In this work, we evaluate the performance of state-of-the-art personalised federated learning algorithms on varying types on non-IID data.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

1 Introduction

In recent years, federated learning (FL) has gained interest due it's ability to leverage many different devices during the training process. The computational power of mobile devices has grown significantly allowing mobile devices with limited resources train machine learning models. Previously, edge devices would transmit data to a centralised server for training. However, this method suffers from privacy problems. With federated learning, each client trains their version of the model on local data and periodically communicates with a server which aggregates updates from the clients to train a centralised model. This helps preserve privacy by communication parameter updates instead of user data. In an ideal case, the global model obtained from aggregating client models should outperform local models due to more data being used. Whilst this has many benefits, it still suffers from some issues pertaining to privacy, fairness and how to deal with heterogeneous data. The aim of traditional FL algorithms such as FedAvg is to train a centralised model that performs well on average. We define f as the loss corresponding to user i.

$$min_{w \in \mathbb{R}} d \; f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

For a supervised task with features x and labels y, we aim to solve:

$$f_i(w) := E_{(x,y) \sim p_i}[l_i(w; x, y)]$$

Due to the diversity of the clients available, it can not be assumed that the data for each client comes from the same distribution. Training a global model with non-independent and identically distributed (non-IID) data may lead to problems such as model parameter divergence and non-guaranteed convergence as different models may provide opposing gradient updates. It is widely known that an increase in statistical diversity leads to a significant increase in generalization error on local client data. Despite this challenge, non-IID data also provides an opportunity to create more personalised FL models for individual clients.

1.1 Problem formulation

It is generally well understood what non-IID means. But data can be non-IID in a variety of ways. Client data may be heterogeneous in different ways pertaining to the distribution of samples or labels across clients. These differences in the types of heterogeneity impacts the success of personalisation methods. We draw an example $(x, y) \sim P_i(x, y)$ from a client's local data. Heterogenous data is defined as when $P_i \neq P_j$ for clients i and j. Kairouz et al. survey ways in which client distributions may be non-IID [1]:

Feature distribution skew - The marginal distributions $P_i(x)$ vary across clients even with shared $P(x \mid y)$.

Label distribution skew - Typically when clients are based in different geographical locations. For example, it may be more common to see snow in one location than another. $P_i(y)$ may differ across clients even with shared $P(x \mid y)$.

Concept drift (same label, different features or vice-versa) - When the same label may correspond to different features or the same features correspond to different labels. Could be due to cultural differences, weather effects etc. For example, road signs could look different in different parts of the world. $P_i(x \mid y)$ varies across clients even with shared $P(y)$ or $P_i(y \mid x)$ varies across clients even with shared $P(x)$.

Quantity skew - When certain clients have significantly more data than others

Despite the fact that real-world data typically contains a combination of these effects, most current works on personalised FL focus only on label distribution skew or quantity skew, i.e when the data comes from an existing, flat dataset which is then partitioned across clients by label and/or quantity. To our knowledge little work has been done to study these effects in isolation in an effort to understand their impact on the effectiveness of personalisation methods.

In this work, we investigate the effectiveness of state-of-the-art personalisation methods on different types of non-IID data. We focus on accuracy, convergence speed and stability of training. We hope that by understanding the impact different types of heterogeneity have on the performance of personalised FL algorithms, more methods can be developed to mitigate their effects. We evaluate the performance of these algorithms on non-IID variations of the MNIST dataset.

2 Related work

2.1 Federated learning

Federated stochastic gradient descent (FedSGD), typically regarded as a baseline, selects a fraction C of clients from which to sample gradient updates which are then averaged [2].

One of the first and most widely used FL algorithm is Federated Averaging (FedAvg) [3]. Federated averaging is an improvement on FedSGD where the local models are started from the same random initialisation and exchange model updates rather than gradient updates. There have been many proposed approaches to achieving personalisation in recent years. The main objective of these is to outperform FedAvg for individual clients.

2.2 Personalised Federated Learning

Traditional federated learning presents many issues pertaining to communication efficiency and heterogeneity both in devices and datasets. The aim of personalised FL is to leverage the advantages gained from the FL framework but personalise the global model for individual clients. For any one task, a client may hold data that is irrelevant to the task and thus negatively affects training. According to [4] focusing exclusively on global accuracy has a detrimental effect on personalisation. To be more practical, personalised federated learning needs to focus on simultaneously developing a personalised model that performs well for most clients, developing an accurate global model, and achieving fast training convergence in the least number of rounds. Personalised FL has been studied widely with approaches including mixture-of-experts [5] and multi-task learning[6].

In this work, we focus on two algorithms implementing different approaches for personalisation. The first of these is pFedMe [7], an algorithm which uses Moreau envelopes as the clients' regularized loss function with the following l_2-norm for each client:

$$f_i(\theta_i) + \frac{\lambda}{2} \parallel \theta_i - w \parallel^2$$

Where θ represents the personalised model for each client i and λ controls the weight of the global model w on the personalised model. In cases where client data is more abundant and useful, it may be more useful to appropriate a smaller significance to the global model. The personalised model can then optimise on local client data without straying too far from the global model. One advantage of pFedMe is the ability to update the global model similar to FedAvg, whilst optimising a personal model in parallel with low complexity. This can be split into a two step problem:

$min_{w\in\mathbb{R}}d \; \{F(w) := \frac{1}{n}\sum_{i=1}^{n} F_i(w)\}$, where $F_i(w) = min_{\theta_i\in\mathbb{R}^d}\{f_i(\theta_i) + \frac{\lambda}{2} \parallel \theta_i - w \parallel^2\}$

The optimised personalised model is then defined as the unique solution to the second part of the problem.

The results for pFedMe also demonstrate that it can achieve SOTA convergence speedup rate with carefully selected values of λ. Whilst larger values of λ increase convergence speed, it may also lead to divergence if too large.

The other approach to this is meta learning which focuses on building robust and adaptable models by training them on a variety of tasks allowing them to solve new tasks with a small amount of data. Applying centralized model-agnostic meta-learning (MAML) was first proposed in [8]. The training of the model allows it to be effective on a range of tasks whilst being easily fine-tuned for a specific task.

Per-FedAvg, the second algorithm, was developed using MAML assuming a limited budget to update a model when new data is observed [9]. Beyond the aim of finding a good model as a starting point for fine-tuning, Per-FedAvg aims to find a global model to use as an initialisation which still performs well after each client performs a final gradient update step to optimize on it's own loss function. Per-FedAvg uses a similar regularising loss function to pFedMe but only optimises the first order approximation of the loss function whereas pFedMe optimises this directly. However, Per-FedAvg requires computing the Hessian matrix which is computationally costly. Some algorithms have been developed to approximate this such as ARUBA [10].

In the comparison of these algorithms conducted by Dinh et al., the heterogeneous data used contained a combination of label skew and quantity skew. The results obtained demonstrated that pFedMe outperforms vanilla FedAvg and Per-FedAvg. Whether these results hold for other types of heterogeneity is an interesting problem.

3 Experiments

The main goal of this work is to evaluate the effectiveness of two SOTA personalised FL algorithms on variations of non-IID data. We evaluate the impact on convergence speed, training stability and test accuracy. The experiments were conducted on the MNIST dataset [1], a handwritten dataset of digits with 10 classes and 70,000 samples. The three algorithms investigated are pFedMe, Per-FedAvg and FedAvg. In the case of pFedMe, we consider both the personalised and the global model.

Experimental Setup: Experimentation was conducted with a similar setup to the one used in pFedMe with the code adapted from the source code of the paper [2] with the same hyperparameters and settings. The MNIST dataset is partitioned amongst $N = 20$ clients. A learning rate of 0.005 was used with a subset of 5 clients. For the personalised version of pFedMe, a local learning rate of 0.1 was used. Larger values of K, the number gradient descent computations, were found to show little improvement on the convergence of both the personalised and global model. As a result, a value of $K = 5$ was chosen. We set λ to 15. For each experiment, the dataset is split with 75% training and 25% test data. Training was run for 20 local epochs with 800 rounds. A l_2-regularised multinomial logistic regression model, the same as the original paper, was also used. Each experiment was repeated 5 times and averaged to obtain more reliable results.

Baseline: A baseline was established by training on IID data from the MNIST dataset. To generate IID data, the MNIST dataset was partitioned into equal parts and distributed amongst clients. This eliminated the aspect of quantity skew, and the pre-shuffled data means the effects of label skew are negligible.

Heterogeneous data: In this work, label skew, quantity skew and concept drift are evaluated separately. Quantity skew was achieved by partitioning the dataset along randomly generated split points. This created a local data size ranging between 216 and 10257 samples. As the dataset was already shuffled, the local data for each client contained a random number of samples from each of the 10 classes. The random seed was set to 1 to ensure reproducibility. We introduce concept drift by asymmetrically altering a label with probability P. The label l_i is changed to label l_{i+1} unless

[1] http://yann.lecun.com/exdb/mnist/

[2] https://github.com/CharlieDinh/pFedMe

$l = 0$ in which case it is changed to 0. Doing so means images with the same features will have different labels. This experiment was conducted with $P = 0.1, 0.25$ and 0.5 to investigate the effects of varying degrees of concept drift. Label distribution skew was achieved by randomly assigning each client 2 classes out of the available 10. However, this introduced an element of quantity skew as each class contained a different number of samples though it can be assumed this effect is negligible as the number of samples per client were of the same order of magnitude.

4 Discussion

Baseline: As expected, training with homogeneous data was very smooth with fast convergence. An accuracy of over 90% was achieved for all algorithms with FedAvg even achieving a higher test accuracy than Per-FedAvg. Any benefit from personalisation is minimal.

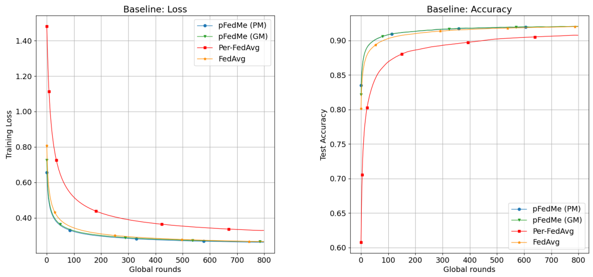

Figure 1: Accuracy and loss over 800 rounds for IID baseline data

Quantity skew: Quantity skew appeared to have very little impact overall. Over 91% accuracy was achieved for all algorithms with smooth training curves and fast convergence as shown in Figure 2. This is comparable to the results obtained with homogeneous data in the baseline experiment. This is likely due to the fact that the algorithms were developed and trained on datasets which contained large amounts of quantity skew. Intuitively, quantity skew is the most likely to occur and therefore the most widely tested form of heterogeneity. This may explain why the personalised FL algorithms are more robust to this kind of statistical heterogeneity .

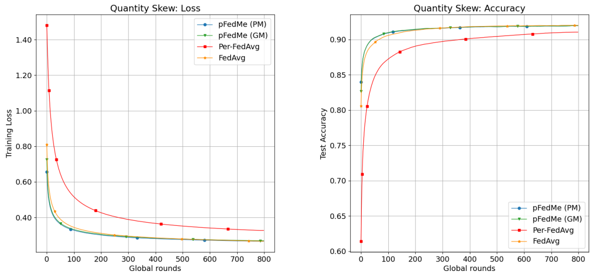

Figure 2: Accuracy and loss over 800 rounds for quantity skew

Label skew: The effects of label skew appear to be quite significant. The accuracy achieved by all algorithms was much lower than in the baseline experiments. However, Per-FedAvg and FedAvg converged faster and appeared much more stable than the two variations of pFedMe. To explain this, we take the example of $client_i$ with samples from classes 0,1. Assuming none of the other clients have samples from class 0 or 1, their models would be of much less relevance to $client_i$. This suggests that less weight should be applied to the global model and therefore a smaller value of λ should be selected to increase the accuracy of the personalised model. To test this hypothesis, we conducted further experiments with 2 other values of lambda. The results show that a value of 15 outperformed both 10 and 20 suggesting that it may actually be the optimal value. Further investigation is necessary to better understand this.

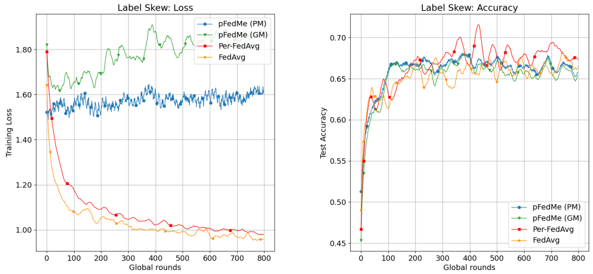

Figure 3: Accuracy and loss over 800 rounds for label skew

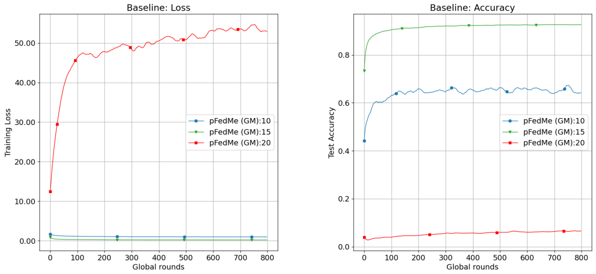

Figure 4: Accuracy and loss over 800 rounds for label skew with $\lambda = 10, 15, 20$

Concept drift: In the case of concept drift with $P = 0.1$, tst accuracy was also impacted slightly though the results were not statistically significant. Th effects of concept drift become more apparent at $P = 0.25$ and 0.5. At $P = 0.25$ the performance of all algorithms begin to suffer, dropping from by almost 20% when compared to the baseline experiment and almost 16% compared to $P = 0.1$. Vanilla FedAvg achieved a slightly higher accuracy than the personalised algorithms. As the concept drift in this case was within local datasets, a higher value of lambda may have improved performance as the input from other clients may have averaged out the effects of the corrupted labels.

An interesting effect is the decrease in accuracy over time. The maximum accuracy for each algorithm is achieved within the first 100 or so rounds before the accuracy begins to decrease though this is only observed for $P = 0.25$ and for Per-FedAvg with $P = 0.5$. It is unclear why this behaviour occurs but a reduced learning rate or introducing dropout may go some way towards mitigating this. Accuracy across all algorithms dropped again when $P = 0.5$ by over 40% compared to the baseline. Training was also much less stable, possibly due to model divergence.

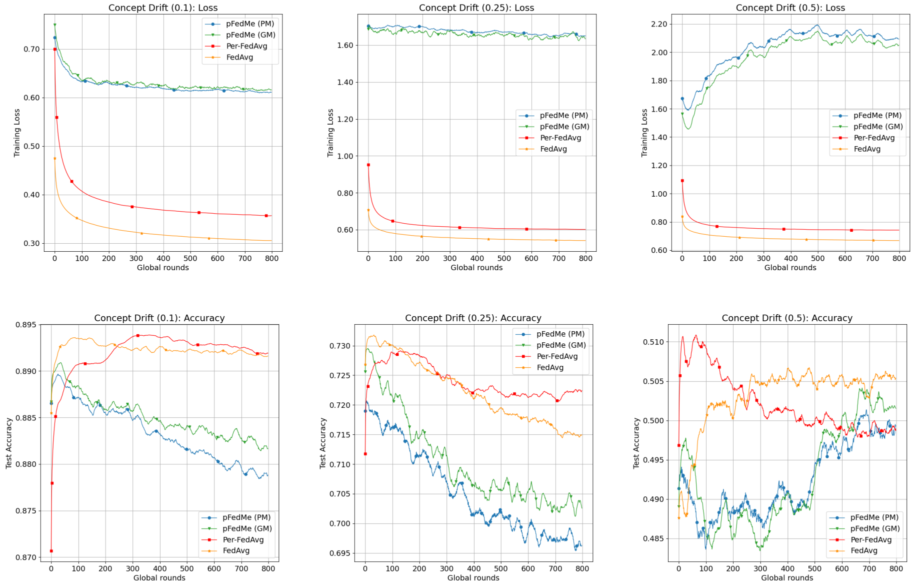

Figure 5: Accuracy and loss over 800 rounds for concept drift with $P = 0.1, 0.25$ and 0.5

Table 1: Max test accuracy obtained for each variation of non-IID data for each algorithm.

				Concept drift		
	Baseline(IID)	Quantity skew	Label skew	0.1	0.25	0.5
pFedMe(GM)	92.08	92.00	68.93	89.15	73.22	50.92
pFedMe(PM)	92.07	92.02	72.44	89.07	72.46	51.01
Per-FedAvg	90.77	91.08	76.00	89.44	73.30	51.57
FedAvg	92.02	92.03	70.33	89.40	73.34	51.16

5 Further work

Due to the long training time for pFedMe (approximately 90 minutes on an NVIDIA Tesla K80 GPU) some alterations had to be made to the methodology. In this work, we repeated the experiments 5 times instead of the recommended 10 which may have affected the reliability of the results. Additionally, the hyperparameters used in this work were obtained from the original experiments but may not have been optimal for the augmented datasets. Experimenting with different hyperparameters may have yielded a better performance, particularly the value of λ used for pFedMe. This work can be extended to include other SOTA personalised FL algorithms such as FedFomo or the combination of FedAvg with Reptile as proposed by [11]. Furthermore, the impact of differential privacy and robust aggregation on the types of non-IID data would also be an interesting extension to this work.

We hope the results of this work can be used to develop methods the mitigate the effects of heterogeneity in personalised FL training algorithms.

6 Conclusion

In this work, we evaluate the performance of pFedMe and per-FedAvg on 3 types of heterogeneous data. The results show that concept drift and label skew impacted the performance of both algorithms more severely than quantity skew. In the case of concept drift, vanilla FedAvg actually outperformed the personalised algorithms. Additionally, the notable difference in the way that they affect training can also be observed. This demonstrates that it is important to apply a more considered approach to mitigate the effects of each kind of heterogeneity, rather than treating all heterogeneous data with a one-fits-all solution.

References

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[2] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, (New York, NY, USA), p. 1310–1321, Association for Computing Machinery, 2015.

[3] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016.

[4] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *CoRR*, vol. abs/1909.12488, 2019.

[5] D. W. Peterson, P. Kanani, and V. J. Marathe, "Private federated learning with domain adaptation," *CoRR*, vol. abs/1912.06733, 2019.

[6] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," *CoRR*, vol. abs/1705.10467, 2017.

[7] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *CoRR*, vol. abs/2006.08848, 2020.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *CoRR*, vol. abs/1703.03400, 2017.

[9] A. Fallah, A. Mokhtari, and A. E. Ozdaglar, "Personalized federated learning: A meta-learning approach," *CoRR*, vol. abs/2002.07948, 2020.

[10] M. Khodak, M. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," *CoRR*, vol. abs/1906.02717, 2019.

[11] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *CoRR*, vol. abs/1803.02999, 2018.