pFedV: Mitigating Feature Distribution Skewness via Personalized Federated Learning with Variational Distribution Constraints

Yongli Mou[1], Jiahui Geng[2], Feng Zhou[3](✉), Oya Beyan[4], Chunming Rong[2], and Stefan Decker[1,5]

[1] Chair of Computer Science 5, RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany
{mou,decker}@dbis.rwth-aachen.de

[2] Faculty of Science and Technology, Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway
{jiahui.geng,chunming.rong}@uis.no

[3] Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China
feng.zhou@ruc.edu.cn

[4] Institute for Medical Informatics, Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, Germany
oya.beyan@uni-koeln.de

[5] Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany
stefan.decker@fit.fraunhofer.de

Abstract. Statistical heterogeneity, especially feature distribution skewness, among the distributed data is a common phenomenon in practice, which is a challenging problem in federated learning that can lead to a degradation in the performance of the aggregated global model. In this paper, we introduce pFedV, a novel approach that leverages a variational inference perspective by incorporating a variational distribution into neural networks. During training, we add the KL-divergence term to the loss function to constrain the output distribution of layers for feature extraction and personalize the final layer of models. The experimental results demonstrate the effectiveness of our approaches in mitigating the distribution shift in feature space in federated learning.

Keywords: federated learning · statistical heterogeneity · variational inference

1 Introduction

Despite the impressive results that deep learning-based approaches have achieved in recent decades, training deep learning models is data-driven and intensively depends on the availability and accessibility of high-quality data. Conventionally,

H. Kashima et al. (Eds.): PAKDD 2023, LNAI 13936, pp. 283–294, 2023.
https://doi.org/10.1007/978-3-031-33377-4_22

data is brought to the computation by following a data centralization approach, leading to privacy breaches and the loss of data sovereignty. As the related issues are increasingly aware, data protection legislation has emerged worldwide in the last few years, e.g., the General Data Protection Regulation (GDPR) in the European Union explicitly prohibits organizations from exchanging data without clear consent from users. Besides, commercial competition and complicated administrative procedures also hinder data integration and data sharing, which makes data exist in the form of isolated islands [22]. As a promising paradigm to provide privacy protection in machine learning, federated learning [16] has been widely adopted in academia and industry. Federated learning enables the participating clients collaboratively train a global machine learning model without revealing local private data. Due to its privacy-preserving characteristics, federated learning is increasingly drawing attention from a wide range of applications and domains such as healthcare [18], finance [22,23], and IoT [8].

Despite federated learning's benefits, its continued popularity is usually accompanied by new emerging problems [6,11], such as the lack of trust among participants, the vulnerability exposed to privacy inferences, the limited or unreliable connectivity, etc. Among these, statistical heterogeneity is considered to be the most challenging problem. It is also called the non-IID problem, where data are not independent and identically distributed across clients. For example, medical radiology images in different hospitals are acquired by different devices using disparate standards [14]. Studies have shown that non-IID data can lead to poor accuracy and slow convergence, sometimes even divergence, if without appropriate optimization algorithms [13]. In practice, the non-IID scenarios are complicated to be categorized, but statistical heterogeneities with regard to label distribution, feature distribution and quantity are mainly being studied. To tackle the aforementioned challenges, it is necessary to adopt appropriate optimization algorithms for federated learning.

In this work, we mainly focus on the feature distribution skewness problem. The main contributions of our paper could be summarized as follows: **(1)** we propose a novel FL training strategy, called pFedV to mitigate the covariance shift, i.e., one of the major problems of statistical heterogeneity. The last layer for feature extraction is modified before the classification layers in the neural networks, instead of compressing the input into the hidden feature space, that layer generates the variational distribution of the feature maps. A regularization term is added in the loss function for the local training in federated learning, i.e., the KL-divergence term makes the variational distribution of the local model close to the output distribution of the global model or a certain pre-defined distribution. We design two variational distribution models, a strong restricted one using zero-mean, unit-variance Gaussian for all clients and another one using the distribution in the global model. **(2)** Furthermore, we adopt the idea of FedBN [14] to train the last classification layer individually at each client, as a personalized technique for federated learning. **(3)** Finally, we evaluate our proposed approaches on five related but heterogeneous data sets and our empirical studies validate pFedV's superior performance on non-IID data.

2 Related Work and Background

2.1 Federated Learning

Unlike conventional machine learning where training is centralized and the data is collected from different sites and stored in central storage [1], federated learning is a distributed machine learning paradigm and trains a global model across data generated from distributed clients participating in each communication round. A typical federated learning system consists of a server and clients, where the server orchestrates the training process by repeating the steps including client selection, model distribution, client training and model aggregation [6], and the clients train the global model with local data. The server aggregates the collected client models according to a specified strategy and the aggregated global model is expected to surpass the performance of independently trained client models. Considering multi-class classification problem, given K clients with client i holding a dataset $\mathcal{D}_i := \{(\mathbf{x}_i^{(n)}, y_i^{(n)})\}_{n=1}^{N_i}$, where $\mathbf{x}_i^{(n)} \in \mathcal{X} \subseteq \mathbb{R}^D$ and $y_i^{(n)} \in \{1, 2, \cdots, C\}$, N_i is the number of data on client i, D is the number of input dimension and C is the number of classes, federated learning can basically be formalized as an optimization problem to minimize the objective function $\min \mathcal{F}(\theta) = \sum_{i=1}^{K} \pi_i \mathcal{F}_i(\theta)$, where θ, π_i and $\mathcal{F}_i$ are the global model, the relative impact and the local objective function $\mathcal{F}_i(\theta) = \frac{1}{N_i} \sum_{n=1}^{n=N_i} \mathcal{L}(\theta, \mathbf{x}_i^{(n)}, y_i^{(n)})$ for client i, respectively. The relative impact π_i can be user-defined with $\sum_{i=1}^{K} \pi_i = 1$ normally as N_i/N, where $N = \sum_{i=1}^{K} N_i$ is the total number of samples. FedSGD [16] used stochastic gradient decent as the optimizer and updated the model on the server for each local training step. However, this approach has a main obstacle i.e., high communication cost. and potential risk of data leakage from gradients [5]. To reduce the communication cost and prevent privacy leakage, FedAvg [16], instead of the one-step gradient descent scheme, is an aggregation strategy that updates models with multiple steps.

2.2 Statistical Heterogeneity

The local objective function $\mathcal{F}_i$ is often defined as the empirical risk over local data and is the same across all clients, while the local data distribution $P_i(X, Y)$ often varies among different clients capturing data heterogeneity. The joint distribution $P_i(X, Y)$ can be rewritten as $P_i(X|Y)P_i(Y)$ and $P_i(Y|X)P_i(X)$ and Kairous et al. simplified the non-identical distributions into five categories, namely (1) covariate shift as feature distribution skew, (2) prior probability shift as label distribution skew, (3) concept shifts including same label- but different feature distributions and same feature- but different label distributions, and (4) quantity skew [6]. In practice, the non-identical distribution can be combined and even more complicated. Studies [13] show that the performance on the convergence rate and the accuracy of FedAvg on heterogeneous data are significantly reduced, compared to the results on homogeneous data. Empirical works address non-IID issues by modifying operations in different steps. For example, FedProx [12] used a proximal term in the local training stage as a regularization term to suppress the divergence of model updates. FedNova [21] improved the aggregation stage by considering different parties may conduct different numbers of local steps. Li et al. [10] proposed comprehensive data partitioning strategies to cover the typical non-IID data cases. To mitigate such performance degradation, FedBN [14] is designed to alleviate the feature shift before averaging models via local batch normalization. Anit et al. [20] chose to add a proximal item to reduce the difference between the global model and the local model parameters, avoiding the failure of convergence during training. Mou et al. [17] demonstrated that additional small balanced datasets can be used to overcome model differences caused by class imbalance. Sai et al. [7] proposed SCAFFOLD that uses a control variable (variance reduction) to correct for client drift in local updates, which is claimed to reduce the number of communication rounds required for training and the impact due to data heterogeneity or client sampling. Recently, a lot of work apply the Bayesian framework to federated learning. Instead of maximizing the log-likelihood $\log p(\mathcal{D}|\theta)$, the Bayesian framework is to find the posterior of model parameters as $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$, where $p(\theta)$ is the prior of model parameters, $p(\mathcal{D}|\theta)$ is the likelihood. FedBE [4] adopted Bayesian inference to achieve robust aggregation of local models through Bayesian model ensemble. It uses Gaussian or Dirichlet distributions and Monte to efficiently model data distributions. FOLA [15] proposed to approximate the client and server posteriors using online Laplacian approximation, and employed a multivariate Gaussian on the server side to construct and maximize the global posterior, thereby reducing aggregation errors and local forgetting due to large model differences. pFedBayes [24] introduced the uncertainty of weights, i.e., Bayesian neural networks (BNNs) [3], into the federated learning system. Each client achieves personalization by balancing between the construction error of its own private data and the KL divergence with the global model.

2.3 Variational Inference

Variational autoencoder (VAE) [9] is a generative model that consists of an encoder yielding approximate posterior distribution $q_\theta(z|x)$ and a decoder yielding approximate likelihood distribution $p_\phi(x|z)$. The objective of VAE is to minimize the KL-divergence between approximate posterior and real posterior as shown in Eq. 1.

$$D_{KL}(q_\theta(z|x)||p(z|x)) = -\int q_\theta(z|x)\log(\frac{p(z|x)}{q_\theta(z|x)})\mathrm{d}z \tag{1}$$

The evidence lower bound (ELBO) is defined as the boxed part on the right-hand side in Eq. 2. We note that the log probability of the data on the left-hand side in Eq. 2 is a constant, therefore maximizing the ELBO is equal to minimizing the KL-divergence.

$$\log p(x) = D_{KL}(q_\theta(z|x)||p(z|x)) + \boxed{\int q_\theta(z|x)\log(\frac{p_\phi(x|z)p(z)}{q_\theta(z|x)})\mathrm{d}z} \tag{2}$$

The ELBO can be derived into two terms, namely, the KL-divergence term and the reconstruction term as shown in Eq. 3. The KL-divergence term is a constraint on the form of the approximate posterior as a regularizer while the reconstruction term is a measure of the likelihood of reconstructed data output at the decoder. The detailed derivation is available in [19].

$$\mathrm{ELBO} = \int q_\theta(z|x)\log(\frac{p(z)}{q_\theta(z|x)})\mathrm{d}z + \int q_\theta(z|x)\log(p_\phi(x|z))\mathrm{d}z \tag{3}$$

$$= D_{KL}(q_\theta(z|x)||p(z)) + \mathbb{E}_{z\sim q_\theta(z|x)[\log p_\phi(x|z)]} \tag{4}$$

3 Methodology

3.1 Problem Formulation

As mentioned above, we consider the horizontal federated learning scenario (i.e., each client shares the same feature space but differs in sample ID space) with a supervised learning task (e.g., multi-class classification). We use neural networks for the task and formalize as a function $f(\mathbf{x}) = h(g(\mathbf{x}))$ consisting of two parts, i.e., $g(\cdot)$ is the encoder function parameterized by θ_g that extracts input features and the $h(\cdot)$ is the classifier function parameterized by θ_h that classifies the extracted features. We write $\mathbf{z} = g(\mathbf{x})$ and $y = h(\mathbf{z})$, where $\mathbf{z} \in \mathbb{R}^M$ and M is the dimension of the latent representations. Usually, deep neural networks are formed by stacking layer upon layer. Therefore, the parameters of the encoder and classifier can be further formulated as $\theta_g = (\theta_g^{(1)}, \theta_g^{(2)}, \cdots, \theta_g^{(G)})$ and $\theta_h = (\theta_h^{(1)}, \theta_h^{(2)}, \cdots, \theta_h^{(H)})$, where G and H are the number of layers in the encoder and classifier, and $\theta_g^{(i)}$ and $\theta_h^{(j)}$ denote the parameters of i-th and j-th layer in the encoder and classifier, respectively. For the statistical heterogeneity, we focus on feature distribution skewness, i.e., for two clients, their corresponding joint distributions vary due to the covariate shift, i.e., $P_i(X,Y) \neq P_j(X,Y), \forall i \neq j$ since $P_i(X) \neq P_j(X), \forall i \neq j$, assuming the conditional distribution $P(Y|X)$ is shared across clients.

3.2 Derivation of Variational Distribution Constraints

In our model, we denote the input and output of the neural networks as $\mathbf{x}$ and y and the latent representation as $\mathbf{z}$. We aim to learn the true posterior distribution $p(\mathbf{z}|y)$ for a given label y, which ensures that the learned latent representation is informative about the label and can be used to make accurate predictions on new data. In general, it is difficult to infer the posterior of latent variable $\mathbf{z}$ for a given label y when the likelihood is non-conjugated to the prior. To circumvent this issue, we resort to the variational inference [2] which uses a variational distribution to approximate the true posterior. Following the standard variational inference, the objective is to minimize the KL divergence between the variational distribution $q_\theta(\mathbf{z})$[1] and the true posterior (as shown in Eq. 5) to learn a variational distribution that is as close as possible to the true posterior, which is equivalent to maximizing the evidence lower bound (ELBO).

$$D_{KL}(q_\theta(\mathbf{z})||p(\mathbf{z}|y)) = -\int q_\theta(\mathbf{z})\log(\frac{p(\mathbf{z}|y)}{q_\theta(\mathbf{z})})\mathrm{d}z \tag{5}$$

Similar to the derivation of the ELBO of VAE, we derive the ELBO[2] as in Eq. 6. Basically, ELBO consists of two parts: on the one hand, it enforces the model to fit the data better with the log-likelihood term; and on the other hand, it makes the variational distribution $q_\theta(\mathbf{z})$ as close as possible to the prior $p(\mathbf{z})$ by using Kull-back-Leibler (KL) divergence.

$$\mathrm{ELBO} = \mathbb{E}_{\mathbf{z}\sim q_\theta(\mathbf{z})}\log p(y|\mathbf{z}) - D_{KL}(q_\theta(\mathbf{z})\|p(\mathbf{z})), \tag{6}$$

Our goal is to find the optimal variational distribution of the latent representation $\mathbf{z}$. Specifically, we assume the variational distribution of $\mathbf{z}$ is a Gaussian distribution $q_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\theta_g}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\theta_g}(\mathbf{x})))$ where $\mathrm{diag}(\cdot)$ denotes the diagonalization of a vector. The mean and variance are modeled by an encoder whose parameters are denoted as θ_g. After drawing a $\mathbf{z}$ from the corresponding approximate variational distribution from $q_\theta(\mathbf{z})$, known as the reparameterization trick [9], we can classify the current sample with the help of a classifier constructed by another neural network $\hat{y} = h(\mathbf{z})$ where $\hat{y}$ is the predicted class label and $h(\cdot)$ denotes the classifier parameterized by θ_h. We replace the log-likelihood term in 6 by the cross entropy loss in our case and finally obtain the following objective for our model, where CE is the cross entropy loss:

$$\theta^*_g, \theta^*_h = \underset{\theta_g,\theta_h}{\mathrm{argmin}}\ \mathbb{E}_{q_{\theta_g}(\mathbf{z})}\mathrm{CE}(\hat{y}_{\theta_h}(\mathbf{z}), y) + \alpha D_{KL}(q_{\theta_g}(\mathbf{z})\|p(\mathbf{z})), \tag{7}$$

Comparing to conventional classification model training, a KL divergence term is added to the objective function as shown above. We add a weight factor α to the KL term, which is a hyperparameter, to adjust the strength of the penalty. In our case, we set it to 0.5 in all experiments related to variational distribution.

3.3 Personalized Federated Learning with Variational Distribution Constraints

In this section, we present our proposed approach, personalized federated learning with variational distribution constraints (pFedV). Figure 1 gives the overview of pFedV.

[1] The variational distribution is the output of the encoder parameterized by θ, which is equivalent to θ_g in the previous section.

[2] We omitted x in the formula since all distributions are given the condition of $\mathbf{x}$, e.g., $q_\theta(\mathbf{z}) = q_\theta(\mathbf{z}|\mathbf{x})$.

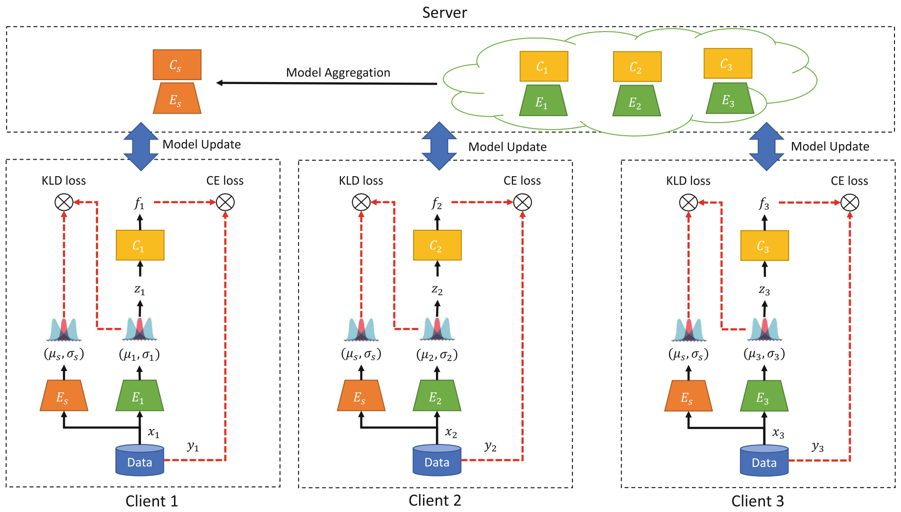

Fig. 1. Overview of pFedV: At each communication round, the server sends the global model to the clients participating in the local training; During the local training, models are trained with the above-mentioned loss function; After training for a given number of epochs, model updates are sent back to the server for the model aggregation, in which the last layer for classification is reserved at each client if the personalized setting is chosen

Like conventional federated learning systems, a server is employed for orchestrating the federated learning process repeating the steps of model update and aggregation. The blue bidirectional arrows between the server and clients indicate the communication for the model update. The server sends the global model to the clients at the beginning of each communication round and the clients send local models to the server after the local training.

The variational distribution constraints and loss functions described above are applied during the local training. We make an assumption for the variational distribution, i.e., the Gaussian distribution. The encoder of the neural network is modified to output the mean and standard deviation (for the non-negativity guarantee of standard deviation, we use log variance instead).

For the construction of the prior, we utilize two different strategies: **(1)** a fixed prior distribution like the classical variational inference and **(2)** continuous update. For the fixed prior solution, we use strong prior constraints, i.e., zero-mean, unit-variance Gaussian distribution for all clients. For the continuous update solution, we abstract the aggregated knowledge into the prior distribution and use the output of the variational distribution of the global model, i.e., the prior is constantly updated as the server communicates with clients in our federated learning framework. Specifically, we assume the prior of $\mathbf{z}$ is also a Gaussian distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\theta_s}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\theta_s}(\mathbf{x})))$. The mean and variance are modelled by another encoder whose parameter is denoted as θ_s.

With the construction of prior and variational posterior by two encoders, the KL divergence term in the loss function makes the posterior of extracted features from clients close to the global one.

Furthermore, the personalized variant of federated learning is proposed for personalizing the global model for each client in the federation to overcome data heterogeneity issues. In our approaches, we propose to reserve the parameters of the last layer of the classifier $\theta_h^{(H)}$ to achieve personalization.

4 Experiments

4.1 Experimental Settings

To evaluate the performance of our proposed approaches in the above methodologies in the non-IID scenario of federated learning. we conducted extensive experiments in comparison with baselines, i.e., single-site training and FedAvg, FedProx, FedBN and FOLA. Additionally, we report the results of the conducted experiments and analyze the effect of variational distribution constraints.

Datasets. To demonstrate the feature distribution skewness problem, we conduct all experiments on Digits-Five dataset, namely MNIST, SVHN, USPS, Synthetic Digits and MNIST-M. They all contain digit images and are for the multi-class classification task. Figure 2 shows some sample images of the Digits-Five dataset, from which we can observe the non-IID phenomenon in feature space, i.e., the digits from different datasets vary considerably.

Model. For all experiments presented in this section, we implement a simple convolutional neural network model for classification with three convolutional layers with 5×5 kernel (the first and the second with 64 channels and the last with 128 channels, each followed by batch normalization, 2×2-max pooling and ReLU activation) and three fully connected layers with batch normalization followed by ReLU activation (the first with 2048 units, the second with 512 units and the last with 10 units a.k.a. logits). In between, the extracted feature maps by convolutional neural networks are flattened into a 6272-dimensional vector. For variational distribution, we doubled the channels of the third convolutional layer, that the first half represents the mean and the second half represents the variance of the encoder output, and by using the reparameterization technique draw the feature maps following corresponding distributions.

Setups. MNIST, SVHN, USPS, Synthetic digits, and MNIST-M consist of the training sets of 60000, 73257, 7291, 479400, and 60000 examples and test sets of 10000, 26032, 2007, 9553, 10000, respectively. In our experiments, we set the quality of data at each client to 7291 and models evaluate models on the original test sets. The image size and the number of channels of images are different from each dataset. We resize all data into the size of 28×28 and the number of channels of input data is set to 3. For single-site training, models are trained for 50 epochs, while in federated settings the number of communication rounds is set to 50 and at each communication round, models are trained for one epoch at each client. All experiments adopt the stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and batch size of 32. For FOLA, the weight factor of prior task loss is set to 0.5 (a.k.a., CSD importance) and same for the weight factor of KL divergence term our proposed pFedV. Since the classes are relatively balanced, accuracy (in percentage) is the only metric we used to measure and compare the performance of models trained in different ways.

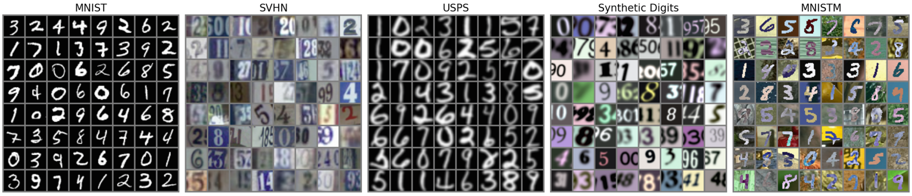

Fig. 2. Example images of datasets used for feature shift (Non-IID) experiments.

4.2 Results

We conduct experiments of single-site training, i.e., models are trained on each client individually and tested on the test sets of MNIST, SVHN, USPS, Synthetic Digits and MNIST-M. The results of the accuracy of single-site trained models are illustrated in Table 1. Each row represents a model trained on the corresponding dataset individually. We can observe that the high-performance values always occur on the diagonal, i.e., models fit well on the test set of the dataset that is the same as that used for training. Of course, there is the possibility of overfitting due to small data sets. We also found that feature complexity is also one of the factors to influence model performance. For example, MNIST and USPS are two datasets with relatively simple features, while SVHN is much more complex as it often occurs more than one digit in one single picture obtained from street view. The interesting result in this table is that the MNIST accuracy of the model trained on MNIST-M is even higher than MNIST-M itself since MNIST-M extended MNIST dataset with randomly extracted patch background. The model trained on MNIST-M has learned the basic features of MNIST with additional generalized feature abstraction and thus works even better on MNIST. However, single-site trained models are overall poor in generalization to other datasets. For example, the second column shows that models trained on other datasets can hardly perform well on SVHN test set, e.g., only 7.95% by the model trained on USPS.

To evaluate the contribution of our approaches to overcoming the non-IID problem in federated learning setting, we compare the results with baselines such

Table 1. Results of models via single site training on test sets of MNIST, SVHN, USPS, Synthetic Digits and MNIST-M

Model (trained on)	MNIST	SVHN	USPS	Synthetic Digits	MNIST-M
MNIST	**98.72**	19.73	28.50	14.92	37.28
SVHN	51.48	**85.18**	64.52	81.43	37.11
USPS	24.41	7.95	**97.11**	23.76	18.60
Synth	82.63	77.97	84.26	**95.04**	54.19
MNIST-M	96.63	30.17	56.05	41.94	**93.62**

as FedAvg, FedProx and FedBN, as well as one of the other Bayesian methods FOLA, as illustrated in Table 2. In general, we can see the effectiveness of variational distribution constraints as the results of FedV that without the personalized layer is also improved on all test sets, which also shows the generalization property of the variational distribution constraints. However, compared with continuously updated prior, the fixed prior does not provide a stable generalization guarantee, for example, it is even worst than FedAvg on SVHN. Overall, our pFedV outperforms others as it achieved 2.36%, 1.05%, 1.74% 1.79% improvement on SVHN, USPS, Synthetic Digits and MNIST-M and slight improvement on MNIST in comparison with FedAvg.

Table 2. Results of methods on test sets of MNIST, SVHN, USPS, Synthetic Digits and MNIST-M in the federated setting

Methods	MNIST	SVHN	USPS	Synthetic Digits	MNIST-M
FedAvg	98.86	83.23	96.16	93.43	90.56
FedProx	98.61	83.36	96.01	93.66	90.59
FedBN	98.67	**86.58**	**97.21**	94.06	91.79
FOLA (CSD 0.5)	98.83	86.46	96.86	94.67	90.50
FedV	98.74	84.80	96.71	94.14	90.93
FedV (Gaussian prior)	98.60	83.04	96.51	93.54	90.46
pFedV	**98.91**	85.99	**97.21**	**95.17**	**92.35**
pFedV (Gaussian prior)	98.86	83.65	**97.21**	94.69	91.74

5 Conclusion

In this paper, we propose a novel federated learning training strategy pFedV to tackle the non-IID problem in federated learning, in particular the covariate shift, a.k.a. feature distribution skewness. Through empirical results, we demonstrate that the proposed approaches vastly improved the federated learning accuracy performance under the scenario of non-IID problem where feature distributions vary across the clients and the results are comparable to state-of-the-art methods like FedBN and FOLA. We have shown the generalization capability of variational distribution in federated learning and the advance of it combined with personalization. For future work, it deserves further investigation of the impact of the combination of multiple variational distribution constraint layers, since the framework is scalable. Besides, it will be interesting to explore more non-IID scenarios and extend to more general settings in addition to feature distribution skewness.

Acknowledgements. This work was supported by the German Ministry for Research and Education (BMBF) projects CORD_MI, POLAR_MI, Leuko-Expert and WestAI (Grant no. No. 01ZZ1911M, 01ZZ1910E, ZMVI1-2520DAT94C and 01IS22094D, respectively), CLARIFY Project (Marie Skłodowska-Curie under Grant no. 860627), and by National Natural Science Foundation of China (NSFC) Project (No. 62106121). This research was supported by Public Computing Cloud, Renmin University of China.

References

1. Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y.: Federated learning review: fundamentals, enabling technologies, and future applications. Inf. Process. Manage. **59**(6), 103061 (2022)
2. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International Conference on Machine Learning, pp. 1613–1622. PMLR (2015)
4. Chen, H.Y., Chao, W.L.: FedBE: Making Bayesian model ensemble applicable to federated learning. arXiv preprint arXiv:2009.01974 (2020)
5. Geng, J., et al.: Towards general deep leakage in federated learning. arXiv preprint arXiv:2110.09074 (2021)
6. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends® Mach. Learn. **14**(1–2), 1–210 (2021)
7. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143. PMLR (2020)
8. Khan, L.U., Saad, W., Han, Z., Hossain, E., Hong, C.S.: Federated learning for internet of things: recent advances, taxonomy, and open challenges. IEEE Commun. Surv. Tutorials **PP**, 1 (2021)
9. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
10. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-IID data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978. IEEE (2022)
11. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Process. Mag. **37**(3), 50–60 (2020)
12. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceed. Mach. Learn. Syst. **2**, 429–450 (2020)

13. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of FedAvg on non-IID data. arXiv preprint arXiv:1907.02189 (2019)
14. Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q.: FedBN: Federated learning on non-IID features via local batch normalization. arXiv preprint arXiv:2102.07623 (2021)
15. Liu, L., Zheng, F., Chen, H., Qi, G.J., Huang, H., Shao, L.: A bayesian federated learning framework with online Laplace approximation. arXiv preprint arXiv:2102.01936 (2021)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, pp. 1273–1282. PMLR (2017)
17. Mou, Y., Geng, J., Welten, S., Rong, C., Decker, S., Beyan, O.: Optimized federated learning on class-biased distributed data sources. In: Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML PKDD 2021. Communications in Computer and Information Science, vol. 1524, pp. 146–158. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93736-2_13
18. Nguyen, D.C., et al.: Federated learning for smart healthcare: a survey. ACM Comput. Surv. (CSUR) **55**(3), 1–37 (2022)
19. Odaibo, S.: Tutorial: Deriving the standard variational autoencoder (VAE) loss function. arXiv preprint arXiv:1907.08956 (2019)
20. Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: On the convergence of federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127 3, 3 (2018)
21. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. Adv. Neural. Inf. Process. Syst. **33**, 7611–7623 (2020)
22. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**(2), 1–19 (2019)
23. Yang, W., Zhang, Y., Ye, K., Li, L., Xu, C.-Z.: FFD: a federated learning based method for credit card fraud detection. In: Chen, K., Seshadri, S., Zhang, L.-J. (eds.) BIGDATA 2019. LNCS, vol. 11514, pp. 18–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23551-2_2
24. Zhang, X., Li, Y., Li, W., Guo, K., Shao, Y.: Personalized federated learning via variational Bayesian inference. In: International Conference on Machine Learning, pp. 26293–26310. PMLR (2022)