Data Source Selection in Federated Learning: A Submodular Optimization Approach

Ruisheng Zhang[1,2], Yansheng Wang[1,2], Zimu Zhou[3], Ziyao Ren[1,2], Yongxin Tong[1,2], and Ke Xu[1,2]

[1] State Key Laboratory of Software Development Environment, Beihang University, China
[2] Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, China
{rszhang, arthur_wang, ziyaoren, yxtong, kexu}@buaa.edu.cn
[3] Singapore Management University, Singapore, Singapore
zimuzhou@smu.edu.sg

Abstract. Federated learning is a new learning paradigm that jointly trains a model from multiple data sources without sharing raw data. For the practical deployment of federated learning, data source selection is compulsory due to the limited communication cost and budget in real-world applications. The necessity of data source selection is further amplified in presence of data heterogeneity among clients. Prior solutions are either low in efficiency with exponential time cost or lack theoretical guarantees. Inspired by the diminishing marginal accuracy phenomenon in federated learning, we study the problem from the perspective of submodular optimization. In this paper, we aim at efficient data source selection with theoretical guarantees. We prove that data source selection in federated learning is a monotone submodular maximization problem and propose FDSS, an efficient algorithm with a constant approximate ratio. Furthermore, we extend FDSS to FDSS-d for dynamic data source selection. Extensive experiments on CIFAR10 and CIFAR100 validate the efficiency and effectiveness of our algorithms.

Keywords: Federated learning · Data source selection · Submodularity

1 Introduction

Federated learning (FL) [3, 13] is an emerging distributed learning paradigm among multiple data sources, where a global model is trained collaboratively without sharing their raw local data. It has been applied in various applications such as cross-hospital medical image classification [13], next-word prediction on smartphones [3,7], information retrieval [10,11], etc. In practice, federated learning often relies on a selective subset of data sources rather than the entire federation. Data source selection, also known as client selection [1], is compulsory due to the massive communication overhead between the data owners (*i.e.* clients) and the server, or simply the budget limit to cover all data sources [2, 3, 13].

The necessity of data source selection also arises from the heterogeneity of data, whose partition may significantly vary in label distribution and data quality [5].

Prior data source selection methods in federated learning fall into two categories. The first category focuses on evaluating every data source from a theoretical perspective such as the Shapley value [5, 8]. These schemes ensure optimal selection, yet at the cost of exponential time complexity, which is prohibitive for practical deployment. The second category exploits heuristics or back-box optimization for selection, which tend to be more efficient, but are prone to low accuracy in case of data heterogeneity. For example, the naive FedAvg [3] adopts a simple random sampling strategy. Others utilize local gradient information [1,2] to approximate the contributions of participants. These solutions lack theoretical guarantees and incur severe performance degradation on heterogeneous data.

In this paper, we aim at efficient data source selection with theoretical guarantees. Our solution is motivated by the empirical observation that the accuracy of deep learning models tends to increase logarithmically with the amount of training samples [9]. Such phenomena inspire us to analyze the data source selection problem in the lens of *monotone submodular maximization* [6]. Our main contributions and results are summarized as follows.

- We theoretically prove that data source selection in federated learning aiming at generalization error minimization can be converted to monotone submodular maximization. To the best of our knowledge, this is the first submodularity analysis directly on the generalization error in federated learning.
- We design an efficient data source selection algorithm called FDSS with an approximate ratio of $1 - \frac{1}{e}$, which can make a better trade-off between accuracy and efficiency. We further propose an extension FDSS-d for data source selection with dynamic participants availability.
- Extensive evaluations on real datasets show that our proposed algorithms outperform the state-of-the-arts [2, 5] in terms of test accuracy and communication rounds on heterogeneous data.

2 Problem Statement

2.1 Data Source Selection in Federated Learning

We consider federated learning of model ω over a federation $\mathcal{F} = \{P_1, P_2, \cdots, P_N\}$ of N data sources, where P_i denotes the i-th data source. P_i holds a set of n_i data samples $X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}\}$. Each X_i is independently and identically drawn from a prior distribution π_i. A common objective of federated learning is to minimize the *expected generalization error* $L_g(\pi, \omega) = \mathbb{E}[L_\pi(\omega) - L_P(\omega)]$ on the joint distribution $\pi = \prod_{i=1}^{N} \pi_i$. $L_P(\omega) = \sum_{i=1}^{N} \frac{n_i}{n} L_{P_i}(\omega)$ represents the overall empirical error and $L_{P_i}(\omega)$ is the local empirical error of data source P_i.

We are interested in selecting a subset $F \subset \mathcal{F}$ for federated learning due to the limited budget to recruit all data sources in real-world applications [1–3, 8]. We quantify the contribution of a subset F by the evaluation function below.

$$g(F) = L_g(\pi, \omega_0) - L_g(\pi, \omega_F) \tag{1}$$

where $\omega_F = \sum_{P_i \in F} \frac{n_i}{n} \omega_i$ is the aggregated model parameter from F, and ω_0 is the initial model parameter *i.e.* when $F = \emptyset$. Given the evaluation function, we now define the data source selection problem in federated learning.

Definition 1. *Given a federation $\mathcal{F}$, an evaluation function g, and a cardinality constraint c, we define the static data source selection problem as:*

$$\max_{F \subset \mathcal{F}} g(F) \quad s.t. |F| \leq c \tag{2}$$

Further assume a time sequence $t = 1, 2, \cdots, T$. Let the available data sources at time t be the subset $\mathcal{F}_t \subset \mathcal{F}$. We can define the dynamic data source selection problem as

$$\max \sum_{F_t \subset \mathcal{F}_t} g(F_t) \quad s.t. |F_t| \leq c, \forall t = t_0, t_1, \cdots, T \tag{3}$$

where F_t represents the selected subset at t.

2.2 Submodularity Analysis of Data Source Selection

To analyze the submodularity of the function g, the key idea is to harness the information-theoretic bound [12] for the expected generalization error in federated learning. Our main claim is the following.

Theorem 1. *If for each data source P_i, $n_i = n$, $\pi_i = \mathcal{N}(\nu, \sigma_i^2 I_d)$ and $\sigma_i^2 - \sigma_j^2 \leq \frac{1}{2}\sigma_j^2$ for all $i \neq j$, then the evaluation function $g(F)$ in Eq. (1) is both monotone and submodular.*

Proof. We first estimate the generalization error using the bounds in [12]:

$$\frac{1}{n} \sum_{P_i \in F} \sum_{j=1}^{n_i} \psi_{i+}^{*-1}(I(x_{i,j}; \omega_F)) \leq L_g(\pi, \omega_F) \leq \frac{1}{n} \sum_{P_i \in F} \sum_{j=1}^{n_i} \psi_{i-}^{*-1}(I(x_{i,j}; \omega_F)) \tag{4}$$

where $I(x_{i,j}; \omega_F)$ refers to the mutual information of model parameter w_F and local data $x_{i,j}$, $\psi_+ : [0, b_+) \to \mathbb{R}$ and $\psi_- : [0, b_-) \to \mathbb{R}$ are convex functions. Based on the assumptions of $n_i = n$ and $\pi_i = \mathcal{N}(\nu, \sigma_i^2 I_d)$, we can get $L_g(\pi, \omega_F) = \sum_{P_i \in F} \frac{2d\sigma_i^2}{k^2 n}$. Next, we prove $g(F)$ is monotone. Let F be a subset of $\mathcal{F}$ and $|F| = k\,(k > 1)$, for any P_j such that $P_j \notin F$:

$$g(F \cup P_j) - g(F) \geq \frac{2d(2k+1)\sigma_{min}^2 - 2dk\sigma_{max}^2}{k(k+1)^2 n} \geq \frac{2d(k+2)\sigma_{max}^2}{3k(k+1)^2 n} \geq 0 \tag{5}$$

where $\sigma_{max}(\sigma_{min})$ denotes the maximum(minimum) across all variances. The third inequality results from the bounded data variance of different data sources. Finally, we prove the submodularity of $g(F)$. Let $\Delta_F^j = g(F \cup P_j) - g(F)$ and $F' = F - P_k (P_k \in F)$,

$$\Delta_{F'}^j - \Delta_F^j = \Big(\sum_{P_i \in F'} \frac{2d(2k-1)\sigma_i^2}{k^2(k-1)^2 n} - \frac{2d\sigma_j^2}{k^2 n} \Big) - \Big(\sum_{P_i \in F} \frac{2d(2k+1)\sigma_i^2}{k^2(k+1)^2 n} - \frac{2d\sigma_j^2}{(k+1)^2 n} \Big)$$

$$\geq \frac{2d(6k^2-2)\sigma_{min}^2}{k^2(k-1)(k+1)^2 n} - \frac{2d(4k+2)\sigma_{max}^2}{k^2(k+1)^2 n} \geq \frac{2k + \frac{2}{3}}{k^2(k-1)(k+1)^2 n} 2d\sigma_{max}^2 \geq 0$$

By the arbitrariness of P_k, we can derive the submodularity of $g(F)$.

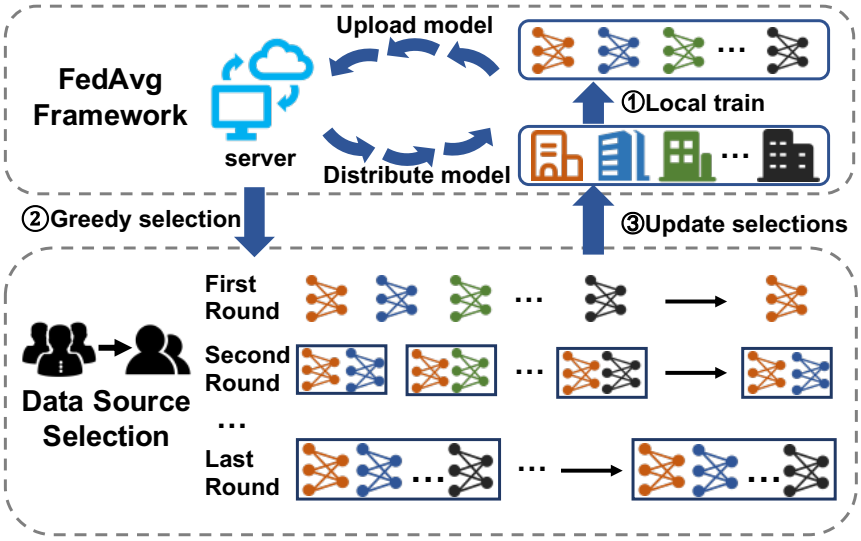


Fig. 1. Illustration of federated learning with data source selection.

3 Data Source Selection Algorithms

Inspired by the submodularity analysis in Sec. 2.2, we devise two greedy data source selection algorithms with constant approximation ratio. The data source selection algorithms can be seamlessly integrated into mainstream federated learning algorithms (see Fig. 1). For ease of presentation, we explain our data source selection algorithms on top of FedAvg [3], but they also function with more advanced federated learning algorithms.

3.1 Static Data Source Selection

As previously mentioned, the monotone submodular maximization nature of the problem ensures a constant approximation ratio by *greedy* selection. To further accelerate the selection process, we also exploit lazy evaluation and approximate the aggregated model. We would like to highlight three aspects of our proposed static algorithm FDSS.

- **Greedy Selection.** The server first initializes ρ as a descending list, and sets the global model and selected federation F to ω_0 and $\emptyset$. Then the algorithm iteratively adds data sources to the federation until c data sources are selected. Let the marginal benefit of g be $\Delta(P_j|F) = g(F \cup P_j) - g(F)$. In each round, the server computes $\Delta(P_j|F)$ and selects the data source that maximizes it, *i.e.*, $F = F \cup \{\arg\max_{P_i \notin F} \Delta(P_i|F)\}$. Afterwards, the server adds the data source to the federation and aggregates global model.
- **Lazy Evaluation.** We use lazy evaluation [4] to accelerate computing the marginal benefit $\Delta(P_j|F)$. Note that the marginal benefit of any data source

P_j is monotonically non-increasing during iteration. Hence, we maintain an upper bound list ρ of $\Delta(P_j|F)$ sorted in descending order. In each iteration, we extract the maximal element P_l from the list ρ and update its benefit. If the updated value is the largest in the current list ρ, submodularity will ensure that the marginal benefits of other data sources are lower than P_j.

– **Approximation of Aggregated Model**. Note that computing $\Delta(P_j|F)$ in each iteration requires calculating $g(F_k \cup P_j)$ for every $P_j \notin F$. However, training a new federated model is time-consuming. For further acceleration, we approximate the federated model by aggregating the trained local model and calculating the accuracy on the global validation set V.

Approximation Ratio and Time Complexity. Assume $F^* = \arg\max_{|F|\leq c} g(F)$ and F_c is the final set in our selection algorithm. According to [6], the greedy selection incurs $g(F_c) \leq (1 - \frac{1}{e})g(F^*)$ theoretically. Let the size of global validation set is m. Then the total time complexity of FDSS without lazy evaluation is $m(N + N - 1 + \cdots + N - c + 1) = c\frac{2N-c+1}{2} = O(N^2)$ if $c = O(N)$. Therefore, the worst-case time complexity of FDSS with the accelerations is $O(N^2)$.

3.2 Dynamic Data Source Selection

Now we extend our FDSS algorithm to the dynamic setting. That is, data source selection is performed in a time sequence $t = 1, 2, \cdots, T$. A naive solution is to repeated perform the static data source selection algorithm *i.e.* FDSS in each round. However, this solution can be inefficient because multiple selections would bring more time cost, especially in scenarios with a large T. And the selected data sources in adjacent rounds are often identical, therefore reselecting data sources is not always necessary. In response, we propose a more efficient dynamic data source selection algorithm FDSS-d. Compared with the naive extension, our FDSS-d algorithm makes the following improvements.

– Each time before calling FDSS, a fast verification is conducted to check whether the global model's accuracy is improving. If the model accuracy is still increasing, there is no need to re-select data sources. In this case, the next selection is postponed till the model converges.
– We divide the entire training into $\frac{T}{s}$ stages and identify the data sources who will participate in the next stage at round ks, where k is a positive integer and s is the selection interval. Note that in the dynamic setting, data sources may be unavailable in each round. Thus we only execute the FDSS algorithm for current online data sources to save bandwidth.

Approximation Ratio and Time Complexity. The FDSS-d algorithm selects data sources dynamically. For each selection, the algorithm guarantees a constant approximation ratio. In the worst case, s selections are executed in total. Hence, the time complexity of FDSS-d is $O(\frac{T}{s}N^2)$. Note that the selection is only performed when the model has converged. Thus the actual running time of FDSS-d is much less than the worst case.

Table 1. Accuracy(%) on CIFAR10. The best performance is marked in bold.

methods	settings				
	Noise 20%	Noise 30%	Noise 40%	Noise 50%	Noise 60%
SFedAvg	26.23 ± 0.96	29.98 ± 0.33	25.86 ± 0.23	33.44 ± 0.29	32.97 ± 0.75
SS-Fed	46.57 ± 0.55	34.94 ± 0.21	28.98 ± 0.82	24.00 ± 0.48	15.60 ± 0.37
FDSS	$\mathbf{51.01 \pm 0.41}$	$\mathbf{53.54 \pm 0.83}$	$\mathbf{52.85 \pm 0.63}$	$\mathbf{44.97 \pm 0.20}$	$\mathbf{50.78 \pm 0.90}$
FedAvg	46.10 ± 1.05	44.46 ± 1.37	41.62 ± 0.82	24.37 ± 0.54	23.38 ± 0.59
Oort	39.32 ± 4.13	34.00 ± 1.66	36.09 ± 2.64	15.53 ± 1.00	34.10 ± 1.86
FDSS-d	$\mathbf{51.38 \pm 0.38}$	$\mathbf{47.98 \pm 0.32}$	$\mathbf{52.52 \pm 0.45}$	$\mathbf{48.60 \pm 0.45}$	$\mathbf{45.70 \pm 0.43}$

Table 2. Accuracy(%) on CIFAR100. The best performance is marked in bold.

methods	settings				
	Noise 20%	Noise 30%	Noise 40%	Noise 50%	Noise 60%
SFedAvg	35.91 ± 0.05	34.82 ± 0.03	35.25 ± 0.05	29.27 ± 0.08	30.66 ± 0.04
SS-Fed	36.67 ± 0.01	35.44 ± 0.06	34.87 ± 0.05	29.45 ± 0.04	31.15 ± 0.06
FDSS	$\mathbf{37.32 \pm 0.01}$	$\mathbf{36.32 \pm 0.05}$	$\mathbf{37.38 \pm 0.05}$	$\mathbf{38.44 \pm 0.06}$	$\mathbf{36.35 \pm 0.08}$
FedAvg	$\mathbf{40.56 \pm 0.06}$	37.69 ± 0.21	35.52 ± 0.23	35.16 ± 0.12	31.95 ± 0.73
Oort	39.29 ± 0.23	36.97 ± 0.24	36.06 ± 0.50	33.14 ± 0.90	32.40 ± 1.43
FDSS-d	40.27 ± 0.07	$\mathbf{39.99 \pm 0.07}$	$\mathbf{38.22 \pm 0.11}$	$\mathbf{39.18 \pm 0.10}$	$\mathbf{37.47 \pm 0.31}$

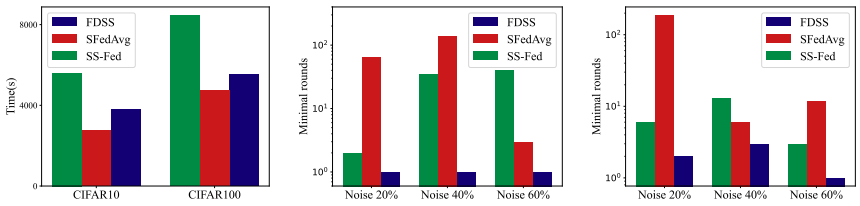
4 Evaluation

4.1 Experiment Settings

Datasets and Models. We compare the performance of different methods on CIFAR10 and CIFAR100. The total number of data sources N is set to 20 and constrained cardinality c is 10. For CIFAR10, We simulate label heterogeneity by allocating data sources with images of different label distributions. The whole dataset has 60,000 images for ten classes. Every data source owns data from two classes randomly. Furthermore, we randomly choose 10 data sources as low data quality enterprises and remap their labels for some training samples [8]. The percentage of noisy data can be used to measure the data quality heterogeneity. For CIFAR100, we make different data sources owning data from the same superclass but different subclasses. We use a two-layer CNN model to recognize images with learning rates 0.02 and 0.01 for CIFAR10 and CIFAR100.

Experimental Environment and Evaluation Metrics. The experiments are conducted on five Intel(R)Xeon(R) Platinum 8269CY 3.10GHz CPUs each with 4 cores. We use test accuracy and training time as evaluation metrics to evaluate model effectiveness and efficiency.

Baselines. We compare our proposed algorithms with the following methods: (1) FedAvg [3], the vanilla Federated Averaging algorithm; (2) SFedAvg, the static version of FedAvg. It randomly selects data sources to participant in FL at the first round; (3) SS-Fed [5, 8], the static version of Shapley-based method. It selects data sources based on Shapley value at the first round; (4) Oort [2], the adaptive selection method based on multi-arm bandit.



(a) Training time (b) Rounds on CIFAR10 (c) Rounds on CIFAR100

Fig. 2. Efficiency evaluation

4.2 Experimental Results

Results on CIFAR10. The results on CIFAR10 are shown in Tab. 1. We evaluate our proposed FDSS and FDSS-d with baselines in five settings, which 20%, 30%, 40%, 50%, 60% of training samples are remapped respectively. The static setting results are shown on the first three rows in Tab. 1. The final accuracy of SFedAvg is rather unstable since it randomly selects data sources at the beginning. We can see that FDSS outperforms baselines in all scenarios. The fourth row shows the accuracy of FedAvg decreases with the increase of noisy data. Compared with FedAvg and Oort, FDSS-d outperforms in all settings and has a minimum accuracy of 45%, which is much larger than baselines.

Results on CIFAR100. The results on CIFAR100 are shown on Tab. 2. For the static version, FDSS performs best as on CIFAR10. The difference is that SFedavg and SS-Fed have a smaller gap with FDSS. The reason may be that each data source has data from all superclasses. Thus the impact of data quality heterogeneity dominates model accuracy. For the dynamical version, FDSS-d outperforms the others in most cases. The only exception is when applying 20% noise, but the difference between FDSS-d and the optimal result is only 0.3%. The results match the previous results and indicate the utility of the proposed algorithm in various heterogeneous scenarios.

Efficiency Evaluation. Fig. 2 shows our efficiency experiments on two datasets. The dynamical algorithms can be seen as calling on static versions repeatedly, so we only show the results of the static algorithms. From Fig. 2a, SS-Fed takes the most time because $c!$ permutations need to compute. Note that the selection is conducted once in the static setting, the efficiency of SS-Fed can be lower for multiple selections. The time cost of FDSS is much less than SS-Fed and close to SFedAvg. We observe from Fig. 2b and Fig. 2c that FDSS has the smallest minimal rounds under all settings.

5 Conclusion

In this paper, we explore data source selection in FL from a submodular optimization perspective. We formalize the data source selection problem in both

the static and dynamic settings and prove that the problem can be converted into a monotone submodular maximization problem. Our theoretical analysis inspires us to devise two greedy-based data source selection algorithms with a constant approximate ratio. Extensive experiments on two real datasets validate the efficiency and effectiveness of our methods.

Acknowledgments. We are grateful to anonymous reviewers for their constructive comments. This work are partially supported by the National Key Research and Development Program of China under Grant No. 2018AAA0101100, the National Science Foundation of China (NSFC) under Grant Nos. U21A20516, 61822201, U1811463 and 62076017, the State Key Laboratory of Software Development Environment Open Funding No. SKLSDE-2020ZX-07, and WeBank Scholars Program.

References

1. Huang, T., Lin, W., Wu, W., He, L., Li, K., Zomaya, A.: An efficiency-boosting client selection scheme for federated learning with fairness guarantee. IEEE Transactions on Parallel and Distributed Systems (2020)
2. Lai, F., Zhu, X., Madhyastha, H.V., Chowdhury, M.: Oort: Efficient federated learning via guided participant selection. In: OSDI. pp. 19–35 (2021)
3. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. pp. 1273–1282 (2017)
4. Minoux, M.: Accelerated greedy algorithms for maximizing submodular set functions. In: Optimization techniques, pp. 234–243. Springer (1978)
5. Nagalapatti, L., Narayanam, R.: Game of gradients: Mitigating irrelevant clients in federated learning. In: AAAI. pp. 9046–9054 (2021)
6. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical programming **14**(1), 265–294 (1978)
7. Shi, Y., Tong, Y., Su, Z., Jiang, D., Zhou, Z., Zhang, W.: Federated topic discovery: A semantic consistent approach. IEEE Intelligent Systems **36**(5), 96–103 (2021)
8. Song, T., Tong, Y., Wei, S.: Profit allocation for federated learning. In: Big Data. pp. 2577–2586 (2019)
9. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV (Oct 2017)
10. Wang, Y., Tong, Y., Shi, D.: Federated latent dirichlet allocation: A local differential privacy based framework. In: AAAI. pp. 6283–6290 (2020)
11. Wang, Y., Tong, Y., Shi, D., Xu, K.: An efficient approach for cross-silo federated learning to rank. In: ICDE. pp. 1128–1139 (2021)
12. Yagli, S., Dytso, A., Poor, H.V.: Information-theoretic bounds on the generalization error and privacy leakage in federated learning. In: SPAWC Workshop. pp. 1–5 (2020)
13. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology **10**(2), 12:1–12:19 (2019)