Choosing Public Datasets for Private Machine Learning via Gradient Subspace Distance

Anonymous Author(s)
Affiliation
Address
email

Abstract

Differentially private stochastic gradient descent privatizes model training by injecting noise into each iteration, where the noise magnitude increases with the number of model parameters. Recent works suggest that we can reduce the noise by leveraging public data for private machine learning, by projecting gradients onto a subspace prescribed by the public data. However, given a choice of public datasets, it is not clear which one may be most appropriate for the private task. We give an algorithm for selecting a public dataset by measuring a low-dimensional subspace distance between gradients of the public and private examples. The computational and privacy cost overhead of our method is minimal. Empirical evaluation suggests that trained model accuracy is monotone in this distance.

1 Introduction

Machine learning models have shown that they can memorize the information of their training data [7]. Recent works have shown that attackers can recover many training samples from published models through carefully designed attacks [3, 18]. This will cause critical privacy issues when the models are trained on private data.

Differential Privacy (DP) [5] is a rigorous privacy criterion that provides theoretical guarantees to the amount of information attackers can infer about any single training point. *Differentially private stochastic gradient descent* (DPSGD) [1, 19, 2] is one of the most popular methods to achieve differential privacy in deep learning. It makes two modifications to vanilla SGD: 1) clipping per-sample gradients to ensure a bound on their ℓ_2 norms; 2) adding Gaussian noise to the gradient.

One downside of applying DP to machine learning is that we need to sacrifice the utility of machine learning models to maintain privacy. Specifically, DPSGD adds random noise drawn from a spherical Gaussian distribution, $\mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbb{I}_p\right)$, where p is the model dimension, i.e., the number of model parameters, and the variance σ^2 scales the noise. The magnitude of the noise introduced in each step scales with the square root of the number of parameters p. For classic deep learning models for Computer Vision tasks like ResNet, the added noise will be tens of times greater than the original gradients, inevitably leading to worse utility.

Many works have proposed various methods to improve the utility of private machine learning [25, 8, 16, 24, 12]. One promising approach involves employing *public* data. Generally, there are two ways of using public data in private training. One is transfer learning, where we pretrain the model on a public dataset and then finetune the model on our target tasks (private data) [16, 1, 23, 15]. Another approach arises from the empirical observation that during the training process, the stochastic gradients always stay in a lower-dimensional subspace of the high-dimensional gradient space p. Based on this observation, some work suggests another approach to leverage public data: they use

Submitted to Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

the public data to find this lower-dimensional subspace and then project the noisy gradient onto this subspace [25, 8, 24, 12]. This generally improves utility over DPSGD without supplementary data.

However, this leaves an open question: *which public dataset should one select for a particular private task?* The ideal case may be if some part of the private dataset is public, as this avoids any distribution shift. But otherwise, we would like a way to quantify a public dataset's fitness for use. Our main contribution is an algorithm for this purpose.

Our method needs a single batch of private and public examples from the dataset. Specifically, the algorithm performs three steps to derive the closeness between public data and private data: 1) compute the per-sample gradient of both public and private data, 2) find the gradient subspace of both private and public data by applying singular value decomposition (SVD), 3) compute the subspace distance d using Projection Metric [11]. Our algorithm gives a value that measures a type of distance between public and private data. Our empirical evaluation shows that the distance d derived from our algorithm follows the utility of the projection method monotonously, meaning that the distance d is a good indicator of public data's utility.

2 Preliminaries

Notation. In this paper, we use p to denote the model dimension, i.e., the number of parameters in the model. k is the dimension of the lower-dimensional space we choose. m refers to the number of examples in a batch. We use superscript or subscript interchangeably to denote private or public data, like x_{priv}, V^{pub}.

Definition 1 (Differential Privacy [5]). *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differential private if for any pair of datasets D, D' that differ in exactly one data point and for all subsets E of outputs, we have:*

$$\Pr[\mathcal{A}(D) \in E] \leq \Pr[\mathcal{A}(D') \in E] + \delta.$$

Definition 2 (Projection Metric [20, 6]). *The projection metric between two k-dimensional subspaces V_1, V_2 is defined as:*

$$d\left(V_1, V_2\right) = \left(\sum_{i=1}^{k} \sin^2 \theta_i\right)^{1/2} = \left(k - \sum_{i=1}^{k} \cos^2 \theta_i\right)^{1/2}$$

where θ_i's come from the principal angles between V_1 and V_2.

Appendix A gives the formal definition of principle angles.

In this paper, we evaluate our method using GEP [24], the state-of-the-art private deep learning algorithm that leverages public data. We briefly describe their algorithm in Appendix A.

3 Methods

Now we define the problem formally. Suppose we have a task that consists of a private dataset X^{priv} and a private deep learning algorithm A that can leverage public data to improve model utility. We have a list of potential choice of public dataset $[X_1^{pub}, X_2^{pub}, \cdots]$. We would like a metric that can prescribe which public dataset, when used with algorithm A on the private task X^{priv}, will have the best model utility.

At a high level, our method involves the following two steps: finding the gradient subspace of the data examples and computing the gradient subspace distance. The algorithm uses the same model A and a batch of unlabeled data examples from private and public datasets. Following standard DPSGD, the algorithm will first compute and store per-example gradients from each data example, that is $\boldsymbol{G_{priv}}, \boldsymbol{G_{pub}} \in \mathbb{R}^{m \times p}$. Then it computes the top-k singular vectors of both the private and public gradient matrix by performing singular value decomposition (SVD). Finally we use projection metric to derive the subspace distance $\boldsymbol{d}$ by taking the right singular vectors $\boldsymbol{V_k^{pub}}$, $\boldsymbol{V_k^{priv}}$ from the previous step. The pseudo-code of our method is given in Algorithm 1.

Our algorithm is based on the empirical observation that the stochastic gradients stay in a lower-dimensional subspace during the training procedure of a deep learning model [10, 14]. We also

Algorithm 1 Gradient Subspace Distance

Input: Private examples x_{priv}, public examples x_{pub}, loss function $\mathcal{L}$, model weights $\mathbf{w}_0$
Output: Distance between two image datasets d

1: $\boldsymbol{G_{priv}} = \nabla\mathcal{L}(\mathbf{w}_0, x_{priv})$ ▷ Compute per-sample gradient matrix for private examples
2: Compute top-k subspace of private gradient matrix $\boldsymbol{V_k^{priv}}$:
$$U^{priv}, S^{priv}, V^{priv} \leftarrow \mathbf{SVD}(G_{priv})$$
3: $\boldsymbol{G_{pub}} = \nabla\mathcal{L}(\mathbf{w}_0, x_{pub})$ ▷ Compute per-sample gradient matrix for public examples
4: Compute top-k subspace of private public matrix $\boldsymbol{V_k^{pub}}$:
$$U^{pub}, S^{pub}, V^{pub} \leftarrow \mathbf{SVD}(G_{pub})$$
5: $\boldsymbol{d} = ProjectionMetric(V_k^{priv}, V_k^{pub})$

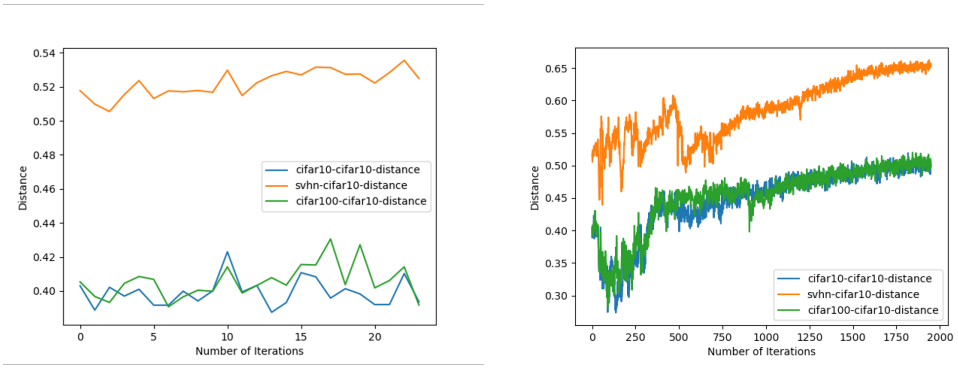
empirically evaluate this finding over different datasets and model settings. Details are given in Appendix C.1. Such observation suggests that most of the information the gradient carries is contented in much lower-dimensional space. Our method finds such subspace for private and public data examples and then measures the distance between two subspaces.

We follow the conclusion in [11] and use projection metric [20, 6] to measure the subspace distance between $\boldsymbol{V_k^{pub}}$ and $\boldsymbol{V_k^{priv}}$. Intuitively, it considers all the principal angles by averaging them to show intermediate characteristics between the two subspaces. It is suggested to be robust to the distribution of data examples and enjoys great distance structure properties such as triangle inequality.

While one may have concern that such distance computation and comparison may have privacy leakage, our method only needs a batch of private examples and use this batch once for distance computation. There will be little privacy leakage during this process. Even if there are some extremely private cases when we have to publish our choices of public data, we can spend some privacy budget and apply some differential privacy mechanism such as exponential mechanism. The scoring function would be the projection metric of $\sqrt{k}$-sensitivity.

4 Experiments

We evaluate our algorithm on three datasets widely used in Computer Vision: Fashion MNIST [22], SVHN [17] and CIFAR-10 [13]. We also choose one medical image dataset: ChestX-ray14 [21], as medical images are considered highly private-sensitive. A variety of datasets are chosen as public data respectively. We use the state-of-the-art private deep learning algorithm that leverage public data, GEP[24], for private training. The details of experiment settings are in Appendix B.



(a) Distance in 1 Epoch (b) Distance over 100 Epoch

Figure 1: The trend of distance during the process of training a Resnet 20 model on CIFAR-10 using vanilla SGD. We follow a standard SGD training procedure and compute the distance between the current private batch and public examples at each iteration.

.

Result. We first empirically evaluate our distance measurement along the training process, as shown in Figure 1. Our empirical study shows that the relative distance between private and public datasets is uniform at most times over the training process. Based on this observation, our algorithm will only require a batch of private examples and one computation step that involves private gradient, meaning the privacy cost and computation overhead is minimal.

Table 1: GEP evaluation accuracy and corresponding distance in descending order. "-" means vanilla DP-SGD training.

Accuracy	Private Dataset	Public Dataset	Distance
58.63%	CIFAR-10	CIFAR-100	**0.20**
57.64%		CIFAR-10	0.24
56.75%		SVHN	0.28
52.16%		-	-
91.32%	SVHN	SVHN	**0.25**
89.29%		CIFAR-100	0.31
89.08%		MNIST-M	0.39
83.21%		-	-
85.25%	FMNIST	FMNIST	**0.34**
84.54%		FLOWER	0.43
83.91%		MNIST	0.50
79.77%		-	-

We compute the distance and evaluate using GEP for the chosen datasets. The evaluation results are given in Table 1. The empirical evaluation shows that the distance derived by our algorithm follows monotonously with the final trained model accuracy. Smaller distance implies that the private examples share more similarities with public examples, thus leading to better accuracy when we use those public data.

Table 2: GEP evaluation AUC and corresponding distance in descending order. "-" means vanilla DP-SGD training.

AUC	Private Dataset	Public Dataset	Distance
69.02%	ChestX-ray14	ChestX-ray14	**0.15**
66.62%		KagChest	0.36
64.90%		-	-
48.80%		CIFAR-100	0.55

For ChestX-ray14, we use the AUC metric because ChestX-ray14 is highly imbalanced where the "no finding" class takes up a large portion of the dataset. The evaluation results are given in Table 2. For more complex tasks, a bad choice of public data, such as CIFAR-100 for ChestX-ray14, will result in worse utility than the DPSGD baseline. When practitioners want to leverage public data for private machine learning, it would be much more essential to use our algorithm to evaluate the quality of the public data before performing private training using algorithms like GEP.

5 Conclusion

While recent studies are focusing on leveraging public data for private machine learning, the quality of public data also matters and is still an open question. In this work, we propose a new algorithm that can help private deep learning practitioners to select public data at minimal time and privacy cost. The empirical evaluation suggests that our distance measurement is a good indicator of public data quality for private machine learning algorithms that leverage public examples.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC*

Conference on Computer and Communications Security, CCS '16, page 308–318, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 464–473. IEEE Computer Society, 2014.

[3] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, August 2021.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[6] Alan Edelman, Tomás A. Arias, and Steven Thomas Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[7] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

[8] Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. *CoRR*, abs/2203.11481, 2022.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations, Third Edition*. Johns Hopkins University Press, 1996.

[10] Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018.

[11] Jihun Ham and Daniel D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 376–383. ACM, 2008.

[12] Peter Kairouz, Mónica Ribero, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private ERM with adagrad rates via publicly estimated subspaces. In *Proceedings of the 34th Annual Conference on Learning Theory*, COLT '21, pages 2717–2746, 2021.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of SGD for deep nets: Dynamics and generalization. In Carlotta Demeniconi and Nitesh V. Chawla, editors, *Proceedings of the 2020 SIAM International Conference on Data Mining, SDM 2020, Cincinnati, Ohio, USA, May 7-9, 2020*, pages 190–198. SIAM, 2020.

[15] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

[16] Zelun Luo, Daniel J. Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5059–5068. Computer Vision Foundation / IEEE, 2021.

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[18] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[19] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013, Austin, TX, USA, December 3-5, 2013*, pages 245–248. IEEE, 2013.

[20] Liwei Wang, Xiao Wang, and Jufu Feng. Subspace distance analysis with application to adaptive bayesian algorithm for face recognition. *Pattern Recognit.*, 39(3):456–464, 2006.

[21] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3462–3471, 2017.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[23] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

[24] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[25] Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private {sgd} with gradient subspace identification. In *International Conference on Learning Representations*, 2021.

A Missing Preliminaries

Definition 3 (Principal Angles [9]). *Let V_1 and V_2 be two orthonormal matrices of $\mathbb{R}^{p\times k}$. The principal angles $0 \leq \theta_1 \leq \cdots \leq \theta_k \leq \pi/2$ between two subspaces span(V_1) and span(V_2), are defined recursively by*

$$cos\theta_k = \max_{\mathbf{u_k}\in span(V_1)} \max_{\mathbf{v_k}\in span(V_2)} \mathbf{u_k'v_k}, \quad \textit{subject to}$$

$$\mathbf{u_k'u_k} = 1, \mathbf{v_k'v_k} = 1, \mathbf{u_k'u_i} = 0, \mathbf{v_k'v_i} = 0, (i = 1, ..., k-1)$$

The first principal angle θ_1 is the smallest angle between all pairs of unit vectors over two subspaces. The rest are similarly defined.

Gradient Embedding Perturbation (GEP). In this paper, we evaluate our method using GEP [24], the state-of-the-art private deep learning algorithm that leverages public data for private training. Here we briefly introduce their algorithm. GEP involves three steps: 1) it computes a set of the orthonormal basis for the lower-dimensional subspace; 2) GEP projects the private gradients to the subspace derived from step 1, thus dividing the private gradients into two parts: embedding gradients that contain most of the information carried by the gradient, and the remainder are called residual gradients; 3) GEP clips two parts of the gradients separately and perturbs them to achieve differential privacy.

B Experiments Setting

Model Architecture. For Fashion MNIST, we use a simple convolutional neural network with around 26000 parameters as in Table 3. For SVHN and CIFAR-10, we use ResNet20 which contains roughly 260,000 parameters. Batch normalization layers are replaced by group normalization layers for different private training, aligning with GEP settings. For ChestX-ray14, we use ResNet152 which has been pretrained on ImageNet1k, a subset of the full ImageNet [4] dataset. We privately fine-tune its classification layer, which contains around 28,000 parameters. We use the same model architecture for subspace distance computation and GEP private training.

Table 3: Model architecture for Fashion MNIST.

Layer	Parameters
Conv2d	16 filters of 8x8, stride=2
Maxpooling2d	stride=2
Conv2d	32 filters 4x4, stride=2
Linear	32 units
Softmax	10 units

Table 4: Choices of public dataset for private dataset. The four datasets in the first row are private datasets. The datasets listed in the first columns are choices of public datasets. 'X' means we choose the two corresponding datasets as a pair of private/public dataset.

	CIFAR-10	SVHN	Fashion MNIST	ChestX-ray14
CIFAR-10	X			
CIFAR-100	X	X		X
SVHN	X	X		
MNIST_M		X		
Fashion MNIST			X	
Flower			X	
MNIST			X	
ChestX-ray14				X
KagChest				X

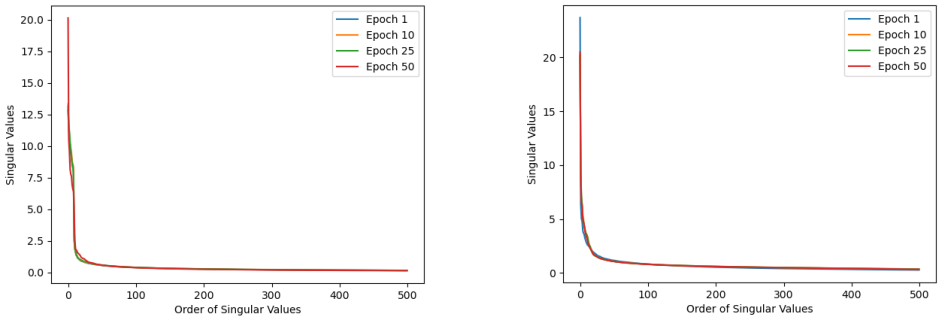
Dataset Choice. ChestX-ray14 consists of frontal view X-ray images with 14 different classes of lung disease. In our evaluation, there are 78,466 training examples and 20433 testing examples in ChestX-ray14. Our choices of public datasets for the four private datasets are described in Table 4. We sample 2000 examples from both private and public datasets for distance comparison using our algorithm. The same 2000 public examples are given to GEP for evaluation.

Hyperparameter Setting. For distance computation, we choose $k = 16$, that is, we only consider a 16-dimensional subspace. We follow the hyperparameter setting in the GEP paper for evaluation. In the GEP paper, they didn't evaluate GEP on the ChestX-ray14 dataset. In our evaluation, we choose $k = 100$ and clip norms are 3 and 1 for original and residual gradients, respectively. The learning rate for the SGD optimizer is set to 0.05. All other hyperparameters are set as default. We use $\epsilon = 2$ and $\delta = 1e - 5$ for all the evaluations.

C More Experiments

C.1 Gradients are in a lower-dimensional subspace.

We evaluate the empirical observation that the stochastic gradients stay in a lower-dimensional subspace during the training procedure of a deep learning model [10, 14], as shown in Figure 2. Results show that only a tiny fraction of singular values are enormous. At the same time, the rest are close to 0, meaning that most of the gradients lie in a lower-dimensional subspace, corresponding to the top singular vectors.



(a) CIFAR-10 (b) ChestX-ray14

Figure 2: Top 500 singular values in the training procedure using vanilla SGD. Model architectures are in the Appendix B. Only a small fraction of singular values are extremely large while the rest are close to 0, meaning that most of the gradients lie in a lower-dimensional subspace, which corresponds to the top singular vectors.

.