PRIVATE LEARNING VIA KNOWLEDGE TRANSFER WITH HIGH-DIMENSIONAL TARGETS

Dominik Fay[⋆†] *Jens Sjölund*[‡] *Tobias J. Oechtering*[⋆]

[⋆] Department of Intelligent Systems, KTH Royal Institute of Technology,
[‡]Department of Information Technology, Uppsala University,
[†] Elekta Instrument

ABSTRACT

Preventing unintentional leakage of information about the training set has high relevance for many machine learning tasks, such as medical image segmentation. While differential privacy (DP) offers mathematically rigorous protection, the high output dimensionality of segmentation tasks prevents the direct application of state-of-the-art algorithms such as Private Aggregation of Teacher Ensembles (PATE). In order to alleviate this problem, we propose to learn dimensionality-reducing transformations to map the prediction target into a bounded lower-dimensional space to reduce the required noise level during the aggregation stage. To this end, we assess the suitability of principal component analysis (PCA) and autoencoders. We conclude that autoencoders are an effective means to reduce the noise in the target variables.

Index Terms— Differential Privacy, Machine Learning, Knowledge Transfer, Image Segmentation, Compression

1. INTRODUCTION

Within differentially private deep learning, two classes of methods are currently predominant. The first one adds noise to the model updates in iterative optimization algorithms, such as noisy stochastic gradient descent [1]. Its primary downside is that the privacy cost scales with the number of model parameters, which ranges in the billions for contemporary language models. The second class of methods is based on noisy knowledge transfer, such as PATE [2, 3, 4], among others [5, 6]. Here, several models are trained on disjoint subsets of the private dataset. Then, these models – referred to as teachers – act as an ensemble to annotate an auxiliary public (but unlabeled) dataset in a privacy-preserving manner. Finally, a student model is trained on this auxiliary dataset. This way, the privacy cost scales with the size of the public dataset instead of the number of model parameters.

The main limitation of PATE is that it is only designed for classification problems, due to the reliance on majority voting at the ensemble stage. A naive extension of PATE to multi-dimensional tasks such as image segmentation could consist of treating the annotation of each individual pixel as a separate classification problem, i.e. majority voting could be performed for each pixel independently. Then, composition theorems could be used to accumulate the privacy cost over all pixels. Unfortunately, the resulting privacy cost would be prohibitively high due to the large number of pixels.

In this paper, we consider multi-dimensional learning tasks with correlated target variables. Since the required noise variance depends on the dimension of the target vector, we propose to map the targets to a lower-dimensional representation so as to decorrelate the targets, before adding the noise. By performing the aggregation and perturbation in this lower-dimensional space, we incur a lower privacy cost than with the naïve setup described above.

The remainder of the paper is organized as follows. In Section 2, we describe our variant of PATE in detail and prove its privacy. We consider PCA as a linear transformation and autoencoders as a nonlinear transformation. In Section 3, we evaluate the performance of the method experimentally. Section 4 concludes with a discussion.

2. METHOD

We begin by describing our PATE variant with a generic compression step and prove its privacy. Then, we describe the linear and non-linear compression methods in detail.

2.1. Problem description

The idea of PATE is to generate a privacy-preserving labeled dataset (for subsequent training) through the use of intermediate teacher models. Each teacher model is a mapping $t_k : \mathcal{X} \to \mathcal{Y}, \mathcal{Y} \subseteq \mathbb{R}^D$ learned from one of K disjoint subsets of the private dataset. All teachers are evaluated on a public unlabeled dataset $\{x_n\}_{n=1..N}, x_n \in \mathcal{X}$ to obtain predictions $y_{nk} = t_k(x_n)$. The predictions are aggregated and perturbed via a randomized mechanism $\mathcal{M} : \mathcal{Y}^K \to \mathcal{Y}$ to obtain $\hat{y}_n = \mathcal{M}(y_{n1}, \ldots, y_{nK})$. The dataset $\{(x_n, \hat{y}_n)\}_{n=1..N}$ gen-

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Data: K teacher models $t_1, \ldots, t_K$; N unlabeled inputs $x_1, \ldots, x_N$; privacy parameters ϵ, δ; encoding and decoding functions h_{enc}, h_{dec}
Result: Student model
Compute $\sigma = \text{AnalyticGauss}\left(\frac{2\sqrt{N}}{K}, \epsilon, \delta\right)$
for *n = 1 to N* **do**
  for *k = 1 to K* **do**
    Run the teacher model $y_{nk} = t_k(x_n)$
    Compress the prediction $z_{nk} = h_{\text{enc}}(y_{nk})$
  end
  Draw $\gamma_n \sim \mathcal{N}(0, \sigma^2 I)$
  Aggregate and perturb $\bar{z}_n = \frac{1}{K}\sum_{k=1}^{K} z_{nk} + \gamma_n$
  Recover the prediction $\hat{y}_n = h_{\text{dec}}(\bar{z}_n)$
end
Train the student model on $\{(x_n, \hat{y}_n)\}_{n=1..N}$

Algorithm 1: Generalized PATE

erated this way is privacy-preserving and is used subsequently to train the student model.

In the context of classification, $\mathcal{M}$ performs noisy majority voting. In this work, on the other hand, we consider mechanisms that consist of three steps: First, they transform the teacher predictions y_{nk} to a low-dimensional representation $z_{nk} \in \mathbb{R}^L$ via a mapping $h_{\text{enc}} : \mathcal{Y} \to \mathbb{R}^L$. Then, they average and perturb the representations by adding noise: $\bar{z}_n = \frac{1}{K}\sum_{k=1}^{K} z_{nk} + \gamma_n$, where $\gamma_n \sim \mathcal{N}(0, \sigma^2 I)$ for some σ^2, and finally transform the representation back into the original space via a reverse mapping $h_{\text{dec}} : \mathbb{R}^L \to \mathcal{Y}$ to obtain $\hat{y}_n = h_{\text{dec}}(\bar{z}_n)$. The above procedure is summarized in Algorithm 1.

In particular, we are interested in mappings $h_{\text{enc}}, h_{\text{dec}}$ that are learned from data in order to fill the low-dimensional space as efficiently as possible. We consider both PCA for its theoretical interpretability and deep neural networks for their ability to detect complex nonlinear patterns. We describe their application in our framework in Sections 2.3 and 2.4, but first we identify the conditions under which Algorithm 1 is differentially private.

2.2. Privacy analysis

The privacy of the algorithm is based on the Analytic Gaussian mechanism [7], described by the following theorem.

Theorem 1 (Analytic Gaussian mechanism [7])**.** *Let f be a function with ℓ_2-sensitivity Δ. For any $\epsilon \geq 0, \delta \in [0, 1]$, if $\mathcal{M}(x) \sim \mathcal{N}(f(x), \sigma^2 I)$ then $\mathcal{M}(x)$ preserves (ϵ, δ)-DP if and only if*

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) - e^{\epsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) \leq \delta, \tag{1}$$

where $\Phi(\cdot)$ is the standard Normal distribution function.

The smallest value for σ such that (1) is satisfied can be found numerically [7]. Let $\text{AnalyticGauss}(\Delta, \epsilon, \delta)$ denote this value. Further, let $dia(S) = \sup\{||u - v|| \mid u, v \in S\}$ denote the diameter of a bounded set. We will now show that Algorithm 1 can be described as a single invocation of the Analytic Gaussian mechanism with appropriate noise variance.

Theorem 2 (Privacy of Alg. 1)**.** *Let h_{enc} be a function with $dia(h_{enc}(\mathcal{Y})) \leq 2$. Then, for any $\epsilon > 0, \delta \in (0, 1]$, Algorithm 1 preserves (ϵ, δ)-DP.*

Proof. Let $\tilde{z} = \text{vec}(\tilde{z}_1, \ldots, \tilde{z}_N)$ denote the stacked vector of noisy target representations. By construction, $\tilde{z}$ is distributed according to

$$\tilde{z} \sim \mathcal{N}\left(f(y_1, \ldots, y_K), \sigma^2 I\right), \tag{2}$$

where

$$f(y_1, \ldots, y_K) = \frac{1}{K}\sum_{k} h_{\text{enc}}(y_k), \tag{3}$$

where we denoted $y_k = \text{vec}(y_{1k}, \ldots, y_{Nk})$ and $h_{\text{enc}}(y_k) = \text{vec}\left(h_{\text{enc}}(y_{1k}), \ldots, h_{\text{enc}}(y_{Nk})\right)$.

Note that any example in the private dataset can only affect the predictions of one teacher, therefore the predictions $(y_k)_{k=1..K}$ of neighboring datasets can only differ at one index k and can only affect one term in (3). W.l.o.g. let this index be K. It follows that f has sensitivity

$$\begin{aligned}
\Delta &= \sup_{y'} ||f(y_1, \ldots, y_K) - f(y_1, \ldots, y'_K)|| && (4)\\
&= \sup_{y'} \frac{1}{K} ||h_{\text{enc}}(y_K) - h_{\text{enc}}(y'_K)|| && (5)\\
&= \sup_{y'} \frac{1}{K} \sqrt{\sum_{n} ||h_{\text{enc}}(y_{nK}) - h_{\text{enc}}(y'_{nK})||^2} && (6)\\
&\leq \frac{2\sqrt{N}}{K}. && (7)
\end{aligned}$$

Since $\sigma = \text{AnalyticGauss}(\Delta, \epsilon, \delta)$, the result follows from Theorem 1. □

2.3. Principal Component Analysis

In this section, we describe the use of PCA as a transformation for Algorithm 1 and derive an analytical criterion for choosing the number of components L to retain.

PCA maps data onto its directions of maximum variance. In this case, the encoder function is given by $h_{\text{enc}}(y) = \pi_L A y$ where the matrix $A = (a_1, \ldots, a_D)^T$ consists of the eigenvectors of the sample covariance matrix and π_L consists of the first L rows of the identity matrix. We assume the eigenvectors to be ordered descendingly according to their eigenvalues. The corresponding decoding is given by $h_{\text{dec}}(z) = A^T \pi_L^T z$.

Assuming a norm bound on the targets, the privacy criterion for Theorem 2 follows immediately by observing that neither A nor π_L can increase the norm.

Theorem 3. *Let* $||y|| \leq 1$ *for all* $y \in \mathcal{Y}$. *Then,* $dia(h_{enc}(\mathcal{Y})) \leq 2$.

Furthermore, we can study the reconstruction error due to perturbation and compression, under the assumption that the marginal label distribution of the teacher ensemble is correct.[1] We consider the expected error over both the data distribution and the randomness of the algorithm. We denote equality in distribution as $\stackrel{d.}{=}$.

Theorem 4. *Let* $\bar{y}$ *be the random variable representing the mean of the teacher predictions and* y *the true target. If* $\bar{y} \stackrel{d.}{=} y$, *then the expected squared error in the reconstruction* $\hat{y}$ *is given by*

$$\mathbb{E}\,||\bar{y} - \hat{y}||^2 = L\sigma^2 + \sum_{\ell > L} \mathbb{E}\left[(a_\ell^T y)^2\right]. \tag{8}$$

Proof. The squared error is given by

$$\begin{aligned} ||\bar{y} - \hat{y}||^2 &= ||\bar{y} - A^T \pi_L^T (\pi_L A \bar{y} + \gamma)||^2 && (9)\\ &= ||(I - \pi_L^T \pi_L) A\bar{y} + \pi_L^T \gamma||^2 && (10)\\ &\stackrel{d.}{=} ||\mathcal{N}\left((I - \pi_L^T \pi_L) A\bar{y},\ \sigma^2 \pi_L^T \pi_L\right)||^2 && (11)\\ &\stackrel{d.}{=} \sum_{\ell \leq L} \mathcal{N}(0, \sigma^2)^2 + \sum_{\ell > L} (a_\ell^T \bar{y})^2, && (12) \end{aligned}$$

Taking expectations yields the result. □

Note that the terms in the left sum in (12) are the error due to noise addition, while each term in the right sum is the variance in the components that are lost due to compression. This illustrates how L controls the trade-off between compression and noise.

We can use this result to derive a criterion for the number of components L to retain. While we do not know $\mathbb{E}\left[(a_\ell^T y)^2\right]$ exactly, we can approximate it by the empirical variance of the training data along the ℓ-th component. Let this variance be denoted by $\hat{\sigma}_\ell^2$. Then, the approximate expected squared error

$$\mathcal{L}_{\text{PCA}} = L\sigma^2 + \sum_{\ell > L} \hat{\sigma}_\ell^2, \tag{13}$$

is minimized by

$$L^* = \underset{\ell \in \{1,\ldots,D\}}{\arg\min} \left\{\hat{\sigma}_\ell^2 | \hat{\sigma}_\ell^2 > \sigma^2\right\}, \tag{14}$$

suggesting that we should keep exactly those components that have a signal-to-noise ratio $\hat{\sigma}_\ell^2 / \sigma^2$ of at least 1.

[1] Note that this is different from assuming a perfect predictor. We are not assuming anything about the predictive distribution $\bar{y}|x$.

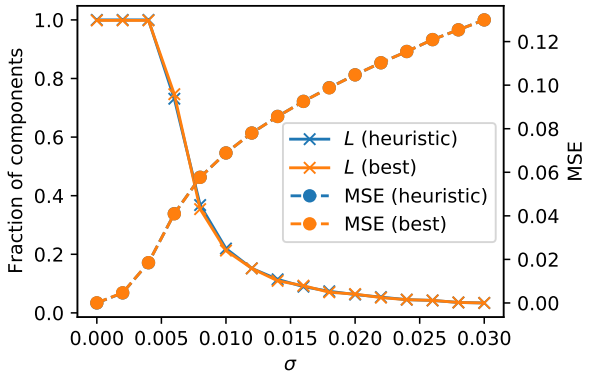

Fig. 1: MSE and optimal fraction of PCA components for varying noise levels. Heuristic refers to (14).

2.4. Autoencoder

An autoencoder is an autoregressive neural network with a low-dimensional representation layer. We denote by $f_\phi : \mathbb{R}^D \to \mathbb{R}^L$ and $g_\psi : \mathbb{R}^L \to \mathbb{R}^D$ the encoder and decoder parts, parameterized by ϕ and ψ, respectively. They are fitted by minimizing a loss function $\ell(x, \hat{x})$ over the reconstructions $\hat{x} = g_\psi(f_\phi(x))$ of the dataset. Hence, a simple way of incorporating autoencoders into our algorithm would be to let $h_{\text{enc}} = f_\phi$ and $h_{\text{dec}} = g_\psi$. However, the resulting representation $f_\phi(y)$ is generally unbounded and would need to be norm-clipped when used in Algorithm 1. Furthermore, we are actually interested in minimizing the loss under the addition of noise. We propose to incorporate both of these factors into the optimization problem by minimizing

$$\mathcal{L}_{\text{AE}}(\phi, \psi) = \mathbb{E}\left[\ell\left(y, g_\psi\left(\frac{f_\phi(y)}{\max\{1, ||f_\phi(y)||\}} + \gamma\right)\right)\right] \tag{15}$$

directly via a stochastic gradient-based method. The resulting transforms for Algorithm 1 are thus $h_{\text{enc}}(y) = \frac{f_\phi(y)}{\max\{1, ||f_\phi(y)||\}}$ and $h_{\text{dec}}(z) = g_\psi(z)$. Since h_{enc} is bounded by construction, the privacy guarantee follows immediately.

3. EXPERIMENTS

We perform a series of experiments on the BraTS 2019 [8, 9, 10] dataset, which consists of preprocessed multi-modal magnetic resonance imaging (MRI) brain scans from 335 subjects, manually labeled with segmentation masks corresponding to the presence of gliomas. The dataset distinguishes between three different regions of the tumor. For simplicity, we consider the binary version of the segmentation task in our experiments, that is, distinguishing the whole tumor from background. Each scan has a resolution of $240 \times 240 \times 155$ voxels, leading to a dimension of $D = 8\,928\,000$. For our experiments, we split the dataset into two training sets of size 130

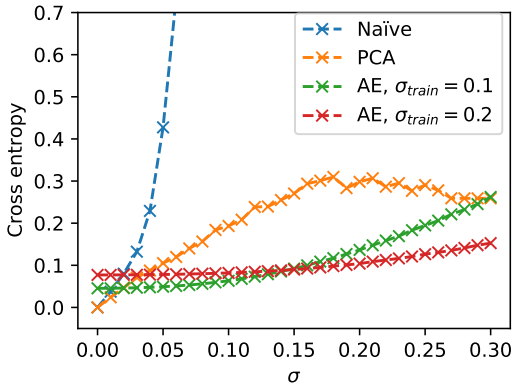

Fig. 2: Cross-entropy between true and noisy segmentation mask for PCA, 3D Autoencoder and Naïve.

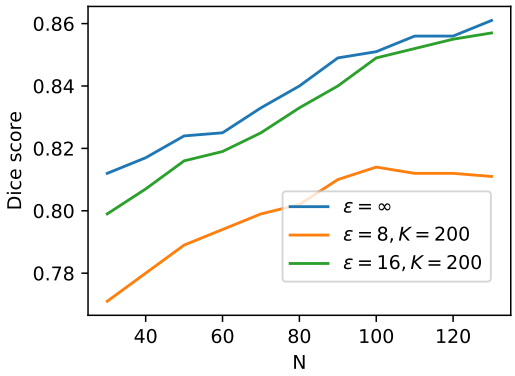

Fig. 3: Dice score of the student model

each, and a test set of size 75. We will refer to these subsets as $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, respectively.

In a first set of experiments (Section 3.1), we train the autoencoder and PCA on the segmentation masks contained in $\mathcal{D}_1$ to assess how the various parameters (σ, L, ϵ, etc.) impact the reconstruction error. The test error is computed on $\mathcal{D}_3$. In a second set of experiments (Section 3.2), we assess the performance of the student model when trained on labels produced by an autoencoder. Since our focus in this work is on learning low-dimensional representations, we do not train any teachers ourselves. Instead, we use the true labels in $\mathcal{D}_2$ as a proxy for the teacher predictions. After encoding, perturbing and decoding them, they form the training set for the student, together with the MR scans of $\mathcal{D}_2$. The test error is computed on $\mathcal{D}_3$.

For PCA, we preprocess the data by splitting each volume into disjoint $24 \times 24 \times 16$ blocks. For 2D Autoencoders, we slice each volume along the third dimension and perform the computation separately on each 240×240 image. All neural networks are trained using the Adam [11] optimizer.

3.1. PCA and Autoencoders

First, we validate the criterion for the optimal number of components L given in (14) experimentally. We do so by comparing it to the empirically optimal number of components which we find by exhaustive search over $\{1, \dots, D\}$. Both numbers are shown for varying noise variances in Figure 1. Furthermore, the resulting mean squared error is shown in both cases.

Next, we compare the reconstruction error of PCA and Autoencoders to the naïve approach of directly addding noise to the segmentation masks in Figure 2. Furthermore, we evaluate how autoencoders generalize to different noise variances than they were trained for. In the figure, σ refers to the noise standard deviation at test time and σ_{train} to that at training time.

3.2. Student training

As the student model we use a 3D U-Net [12]. Figure 3 shows the test error of the student model in relation to the number of annotated examples N. For a fixed privacy cost, increasing N also increases the noise level σ, i.e. there is a trade-off between data quality and data abundance. $\epsilon = \infty$ refers to the non-private baseline, i.e. $N = 130$ without the addition of noise.

4. CONCLUSION

We have proposed a generalization of the PATE framework to make it amenable to high-dimensional learning problems. Due to the versatility of the framework, there are many applications it could find use for if a suitable transform can be found. As examples of such transforms, we have considered PCA, for which we have identified and validated a criterion for controlling the trade-off between compression and perturbation; and autoencoders, by incorporating the privacy-preserving perturbation into the objective function.

Our approach to compressing labels can be seen as an analogous effort to the use of gradient compression in Noisy SGD, which has been used to improve the privacy-utility tradeoff in federated learning [13, 14]. In a federated context (i.e. all teachers are trained at different locations), PATE has the additional advantage of not requiring synchronous communication between clients, which is in contrast to most gradient-based methods [15].

The main criterion for the applicability of our method is the availability of public data. In comparison to methods based on Noisy SGD, an additional unlabeled public dataset is required for the knowledge transfer step.

5. 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*, 2016, pp. 308–318.

[2] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *International Conference on Learning Representations (ICLR)*, 2017.

[3] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson, "Scalable private learning with PATE," in *International Conference on Learning Representations (ICLR)*, 2018.

[4] Sara Saeidian, Giulia Cervia, Tobias J. Oechtering, and Mikael Skoglund, "Quantifying membership privacy via information leakage," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3096–3108, 2021.

[5] Raef Bassily, Abhradeep Thakurta, and Om Dipakbhai Thakkar, "Model-agnostic private learning," in *NeurIPS*, 2018.

[6] Chong Liu, Yuqing Zhu, Kamalika Chaudhuri, and Yu-Xiang Wang, "Revisiting model-agnostic private learning: Faster rates and active learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 838–846.

[7] Borja Balle and Yu-Xiang Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 394–403, PMLR.

[8] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al., "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[9] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, pp. 170117, 2017.

[10] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," 2018.

[11] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[13] Prathamesh Mayekar and Himanshu Tyagi, "Limits on gradient compression for stochastic optimization," *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2658–2663, 2020.

[14] Tran Thi Phuong and Le Trieu Phong, "Distributed sgd with flexible gradient compression," *IEEE Access*, vol. 8, pp. 64707–64717, 2020.

[15] Farhad Farokhi, Nan Wu, David Smith, and Mohamed Ali Kâafar, "The cost of privacy in asynchronous differentially-private machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2118–2129, 2021.