Supplementary material
Differentially Private Sliced Wasserstein Distance

8. Appendix

8.1. Lemma 1 and its proof

Lemma 1. *Assume that $\mathbf{z} \in \mathbb{R}^d$ is a unit-norm vector and $\mathbf{u} \in \mathbb{R}^d$ a vector where each entry is drawn independently from $\mathcal{N}(0, \sigma_u^2)$. Then*

$$Y \doteq \left(\mathbf{z}^\top \frac{\mathbf{u}}{\|\mathbf{u}\|_2}\right)^2 \sim B(1/2, (d-1)/2)$$

where $B(\alpha, \beta)$ is the beta distribution of parameters α, β.

Proof. At first, consider a vector of unit-length in $\mathbb{R}^d$, say $\mathbf{e}_1$, that can be completed to an orthogonal basis. A change of basis from the canonical one does not change the length of a vector as the transformation is orthogonal. Thus the distribution of

$$\frac{(\mathbf{e}_1^\top \mathbf{u})^2}{\|\mathbf{u}\|_2^2} = \frac{(\mathbf{e}_1^\top \mathbf{u})^2}{\sum_i^d u_i^2}$$

does not depend on $\mathbf{e}_1$. $\mathbf{e}_1$ can be either the vector $(1, 0, \cdots, 0)$ in $\mathbb{R}^d$ or $\mathbf{z}$ (as $\mathbf{z}$ is an unit-norm vector). However, for simplicity, let us consider $\mathbf{e}_1$ as $(1, 0, \cdots, 0)$, we thus have

$$\frac{(\mathbf{e}_1^\top \mathbf{u})^2}{\|\mathbf{u}\|_2^2} = \frac{u_1^2}{\sum_i^d u_i^2}$$

where the u_i are iid from a normal distribution of standard deviation σ_u. Hence, because u_1 and the $\{u_i\}_{i=2}^d$ are independent, the above distribution is equal to the one of

$$\frac{\sigma_u^2 V}{\sigma_u^2 V + \sigma_u^2 Z}$$

where $V = u_1^2/\sigma_u^2 \sim \Gamma(1/2)$ (and is a chi-square distribution)) and $Z = (\sum_{i=2}^d u_i^2)/\sigma_u^2 \sim \Gamma((d-1)/2)$ and thus $V/(V+Z)$ follows a beta distribution $B(1/2, (d-1)/2)$. And the fact that $\mathbf{z}$ is also a unit-norm vector concludes the proof.

□

A simulation of the random Y and resulting histogram is depicted in Figure 7.

Remark 3. *From the properties of the beta distribution, expectation and variances are given by*

$$\mathbb{E}Y = \frac{1}{d} \quad \textit{and} \quad \mathbb{V}Y = \frac{2(d-1)}{d^2(d+2)}$$

Remark 4. *Note that if $\mathbf{z}$ is not of unit-length then Y follows $\|\mathbf{z}\| B(1/2, (d-1)/2)$*

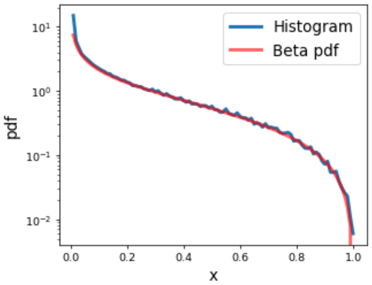

Figure 7. Estimation of the pdf of Y in Lemma 1, for a fixed $\mathbf{z}$, based on a histogram over 100000 samples of $\mathbf{u}$. Here, we have $d = 5$.

8.2. Lemma 2 and its proof

Lemma 2. *Suppose again that $\mathbf{z}$ is unit norm. With probability at least $1 - \delta$, we have*

$$\|\mathbf{XU} - \mathbf{X}'\mathbf{U}\|_F^2 \leq w(k, \delta), \tag{9}$$

with

$$w(k, \delta) \doteq \frac{k}{d} + \frac{2}{3}\ln\frac{1}{\delta} + \frac{2}{d}\sqrt{k\frac{d-1}{d+2}\ln\frac{1}{\delta}} \tag{10}$$

Proof. First observe that:

$$\begin{aligned} H &\doteq \|\mathbf{XU} - \mathbf{X}'\mathbf{U}\|_F^2 = \|(\mathbf{X} - \mathbf{X}')\mathbf{U}\|_F^2 \\ &= \left\|\mathbf{z}^\top \mathbf{U}\right\|_2^2 = \sum_{j=1}^k \left(\mathbf{z}^\top \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|}\right)^2 \\ &= \sum_{j=1}^k Y_j, \text{ where } Y_j \doteq \left(\mathbf{z}^\top \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|}\right)^2. \end{aligned}$$

Therefore, H is the sum of k iid $B(1/2, (d-1)/2)$-distributed random variables.

It is thus possible to use any inequality bounding H from its mean to state a highly probable interval for H. We here use inequality, that is tighter than Hoeffding inequality, whenever some knowledge is provided on the variance of the random variables considered. Recall that it states that if $Y_1, \ldots, Y_k$ and zero-mean independent RV with such that $|Y_i| \leq M$ a.s:

$$\mathbf{P}\left(\sum_{j=1}^k Y_j \geq t\right) \leq \exp\left(-\frac{t^2}{2\sum_{j=1}^k \mathbf{E}Y_j^2 + \frac{2}{3}Mt}\right)$$

For H, we have

$$\mathbf{E}H = \sum_{j=1}^k \mathbf{E}\left(\mathbf{z}^\top \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|}\right)^2 = \sum_{j=1}^k \frac{1}{d} = \frac{k}{d}$$

and Bernstein's inequality gives

$$\mathbf{P}\left(H \geq \frac{k}{d} + t\right) \leq \exp\left(-\frac{t^2}{2kv_d + \frac{2}{3}t}\right),$$

where

$$v_d = \frac{2(d-1)}{d^2(d+2)}$$

is the variance of each $(\mathbf{z}^\top \mathbf{u}_j / \|\mathbf{u}_j\|)^2$ beta distributed variable. Making the right hand side be equal to δ, solving the second-order equation for t give that, with probability at least $1-\delta$

$$H \leq \frac{k}{d} + \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{2kv_d \ln\frac{1}{\delta}}$$

The proof follows directly from Lemma 1 and the fact □

From the above lemma, we have a probabilistic bound on the sensitivity of the random direction projection and SWD . The lower this bound is the better it is, as less noise needed for achieving a certain (ε, δ)-DP. Interestingly, the first and last terms in this bound have an inverse dependency on the **dimension**. Hence, if the dimension of space in which the DP-SWD has to be chosen, for instance, when considering latent representation, a practical compromise has to be performed between a smaller bound and a better estimation. Also remark that if $k < d$, the bound is mostly dominated by the term $\log(1/\delta)$. Compared to other random-projection bounds (Tu) which have a linear dependency in k. For our bound, dimension also help in mitigating this dependency.

8.3. Proof of the Central Limit Theorem based bound

Proof. Proof with the Central Limit Theorem According to the Central Limit Theorem — whenever $k > 30$ is the accepted rule of thumb — we may consider that

$$\frac{H}{k} \sim \mathcal{N}\left(\frac{1}{d}, \frac{v_d}{k}\right)$$

i.e.

$$\left(\frac{H}{k} - \frac{1}{d}\right)\sqrt{\frac{k}{v_d}} \sim \mathcal{N}(0,1)$$

and thus

$$\mathbf{P}\left(\left(\frac{H}{k} - \frac{1}{d}\right)\sqrt{\frac{k}{v_d}} \geq t\right) \leq 1 - \Phi(t)$$

Setting $1 - \Phi(t) = \delta$ gives $t = \Phi^{-1}(1-\delta) \doteq z_{1-\delta}$, and thus with probability at least $1-\delta$

$$\begin{aligned} H &\leq \frac{k}{d} + z_{1-\delta}\sqrt{kv_d} \\ &= \frac{k}{d} + \frac{z_{1-\delta}}{d}\sqrt{\frac{2k(d-1)}{d+2}} \end{aligned}$$

□

8.4. Proof of Property 2.

Property 2. *$DP_\sigma SWD_q^q(\mu,\nu)$ is symmetric and satisfies the triangle inequality for $q = 1$.*

Proof. The symmetry trivially comes from the definition of $\mathrm{DP}_\sigma\mathrm{SWD}_q^q(\mu,\nu)$ that is

$$\mathrm{DP}_\sigma\mathrm{SWD}_q^q(\mu,\nu) = \mathbf{E}_{\mathbf{u}\sim\mathbb{S}^{d-1}} W_q^q(\mathcal{R}_\mathbf{u}\mu * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\nu * \mathcal{N}_\sigma)$$

and the fact the Wasserstein distance is itself symmetric.

Regarding the triangle inequality for $q \geq 1$, our result is based on a very recent result showing that the smoothed Wasserstein for $q \geq 1$ is also a metric (Nietert et al., 2021) (Our proof is indeed valid for $q \geq 1$, as this recent result generalizes the one of (Goldfeld & Greenewald, 2020)). Hence, we have

$$\begin{aligned} \mathrm{DP}_\sigma\mathrm{SWD}_q(\mu,\nu) &= \Big[\mathbf{E}_{\mathbf{u}\sim\mathbb{S}^{d-1}} W_q^q(\mathcal{R}_\mathbf{u}\mu * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\nu * \mathcal{N}_\sigma)\Big]^{1/q} \\ &\leq \Big[\mathbf{E}_{\mathbf{u}\sim\mathbb{S}^{d-1}}\big(W_q(\mathcal{R}_\mathbf{u}\mu * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\xi * \mathcal{N}_\sigma) \\ &\qquad + W_q(\mathcal{R}_\mathbf{u}\xi * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\nu * \mathcal{N}_\sigma)\big)^q\Big]^{1/q} \\ &\leq \Big[\mathbf{E}_{\mathbf{u}\sim\mathbb{S}^{d-1}} W_q^q(\mathcal{R}_\mathbf{u}\mu * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\xi * \mathcal{N}_\sigma)\Big]^{1/q} \\ &\quad + \Big[\mathbf{E}_{\mathbf{u}\sim\mathbb{S}^{d-1}} W_q^q(\mathcal{R}_\mathbf{u}\xi * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\nu * \mathcal{N}_\sigma)\Big]^{1/q} \\ &\leq \mathrm{DP}_\sigma\mathrm{SWD}_q(\mu,\xi) + \mathrm{DP}_\sigma\mathrm{SWD}_q(\xi,\nu) \end{aligned}$$

where the first inequality comes from the fact that the smoothed Wassertein distance $W_q(\mu*\mathcal{N}_\sigma, \nu*\mathcal{N}_\sigma)$ is a metric and satisfies the triangle inequality and the second one follows from the application of the Minkowski inequality.

□

8.5. Experimental set-up

8.5.1. Dataset details

We have considered 3 families of domain adaptation problems based on Digits, VisDA, Office-31. For all these datasets, we have considered the natural train/test number of examples.

For the digits problem, we have used the MNIST and the USPS datasets. For MNIST-USPS and USPS-MNIST, we have respectively used 60000-7438, 7438-10000 samples. The VisDA 2017 problem is a 12-class classification problem with source and target domains being simulated and real images. The Office-31 is an object categorization problem involving 31 classes with a total of 4652 samples. There exists 3 domains in the problem based on the source of the images : Amazon (A), DSLR (D) and WebCam (W). We have considered all possible pairwise source-target domains.

For the VisDA and Office datasets, we have considered Imagenet pre-trained ResNet-50 features and our feature extractor (which is a fully-connected feedforword networks) aims at adapting those features. We have used pre-trained features freely available at `https://github.com/jindongwang/transferlearning/blob/master/data/dataset.md`.

8.5.2. Architecture details for domain adaptations

Digits For the MNIST-USPS problem, the architecture of our feature extractor is composed of the two CNN layers with 32 and 20 filters of size 5×5. The feature extractor uses a ReLU activation function a max pooling at the first layer and a sigmoid activation function at the second one. For the classification head, we have used a 2-layer fully connected networks as a classifier with 100 and 10 units.

VisDA For the VisDA dataset, we have considered pre-trained 2048 features obtained from a ResNet-50 followed by 2 fully connected networks with 100 units and ReLU activations. The latent space is thus of dimension 100. Discriminators and classifiers are also a 2 layer fully connected networks with 100 and respectively 1 and “number of class” units.

Office 31 For the Office dataset, we have considered pre-trained 2048 features obtained from a ResNet-50 followed by two fully connected networks with output of 100 and 50 units and ReLU activations. The latent space is thus of dimension 50. Discriminators and classifiers are also a 2 layer fully connected networks with 50 and respectively 1 and “number of class” units.

For Digits, VisDA and Office 31 problems, all models have been trained using Adam with learning rate validated on the non-private model.

8.5.3. Architecture details for generative modelling.

For the MNIST, FashionMNIST generative modelling problems, we have used the implementation of MERF available at `https://github.com/frhrdr/dp-merf` and plugged in our DP$_\sigma$SWD distance. The generator architecture we used is the same as theirs and detailed in Table 3. The optimizer is an Adam optimizer with the default 0.0001 learning rate. The code dimension is 10 and is concatenated with the one-hot encoding of the 10 class label, leading to an overall input distribution of 20.

For the CelebA generative modelling, we used the implementation of Nguyen et al. (2021) available at `https://github.com/VinAIResearch/DSW`. The generator mixes transpose convolution and batch normalization as described in Table 5. The optimizer is an Adam optimizer with a learning rate of 0.0005. Again, we have just plugged in our DP$_\sigma$SWD distance.

Table 3. Description of the generator for the MNIST and FashionMNIST dataset.

Module	Parameters
FC	20 - 200
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
FC	200 - 784
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
Reshape	28 x 28
upsampling	factor = 2
Convolution	5 x 5 + ReLU
Upsampling	factor = 2
Convolution	5 x 5 + Sigmoid

Table 4. Model hyperparameters and privacy for achieving a $\varepsilon - \delta$ privacy with $\varepsilon = 10$ and δ depending on the size of the private dataset. The four first lines refers to the domain adaptation problems and the data to protect is the private one. The last two rows refer to the generative modelling problems. The noise σ has been obtained using the RDP based moment accountant of Xiang (2020).

data	δ	d	k	N	#epoch	batch size	σ
U-M	10^{-5}	784	200	10000	100	128	4.74
M-U	10^{-5}	784	200	7438	100	128	5.34
VisDA	10^{-5}	100	1000	55387	50	128	6.40
Office	10^{-3}	50	100	497	50	32	8.05
MNIST (b)	10^{-5}	784	1000	60000	100	100	2.94
MNIST (c)	10^{-5}	784	1000	60000	100	100	0.84
CelebA (b)	10^{-6}	8192	2000	162K	100	256	2.392
CelebA (c)	10^{-6}	8192	2000	162K	100	256	0.37

Table 5. Description of the generator for the CelebA dataset. The input code is of size 32 and the output is $64 \times 64 \times 3$.

Module	Parameters
Transpose Convolution	32 - 512, kernel = 4x4, stride = 1
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
ReLU	
Transpose Convolution	512 - 256, kernel = 4x4, stride = 1
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
ReLU	
Transpose Convolution	256 - 128, kernel = 4x4, stride = 1
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
ReLU	
Transpose Convolution	128 - 64, kernel = 4x4, stride = 1
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
ReLU	
Transpose Convolution	64 - 3, kernel = 4x4, stride = 1
BatchNorm	$\epsilon = 10^{-5}$, momentum=0.1
Tanh	