University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
Mathematics of Operations Research

Msc. Thesis

Wasserstein Generative Adversarial Privacy Networks

Kars Mulder

July 19th, 2019

ADVISOR
Dr. ir. Jasper Goseling

GRADUATION COMMITTEE
Dr. ir. Jasper Goseling
Prof. dr. A.J. Schmidt-Hieber
Dr. A. Skopalik

Abstract

A method to filter private data from public data using generative adversarial networks has been introduced in an article “Generative Adversarial Privacy” by Chong Huang et al. in 2018 [1]. We attempt to reproduce their results, and build further upon their work by introducing a new variant based on Wasserstein generative adversarial networks. For certain classes of probability distributions, we prove theorems relating the 1-Wasserstein distance to the amount of private data leaked, and provide counterexamples showing that this relation is not trivial.

Contents

1. Introduction

1.1. Motivation

In the digital age, privacy is becoming an increasingly important societal subject. Storage has become cheap and companies want ever increasing amounts of data about citizens. At the same time, societal awareness about the privacy impact thereof is increasing and laws are getting stricter, for example the GDPR [2]. In order to strike a balance between these conflicting interests, we need mathematical tools to optimize the usefulness of data to companies while minimising the privacy impact of said data.

Not all data is created equal: people are picky about which information they're willing to share and which information they're not. For example, many people talk about their pets on Facebook, but far fewer talk about what medical issues they have. Different levels of sensitiveness of information is also encoded in law: personal information has stronger protection than non-personal information. Medical information frequently enjoys particularly strong protections, such as with the HIPAA law [3].

An intuitive compromise could be "share unsensitive information, do not share sensitive information". However, this runs into a problem: seemingly not sensitive information may correlate with sensitive information, and sharing such seemingly not sensitive information may end up leaking sensitive information anyway.

As an example, consider table 1.1, which shows a fictive dataset that a hospital may have on its patients. The hospital wants to share some of its demographic data with a third party so it can do some helpful analysis. However, medical information such as the disease a person has been diagnosed with is considered to be highly private and must not be shared under any circumstance. As such, this information needs to be filtered before it can be shared.

Simple and intuitive measures to improve privacy could be removing the patient's name and diagnosed disease from the dataset. However, this approach solution isn't perfect. First, there is identification risk: even if the patient's name is removed from the table, a third party may still be able to interfere it based on there being only one person matching

Name	Gender	Age	Zip Code	Disease
Alice	Female	24	34290	Pneumonia
Bob	Male	51	98343	Heart Disease
Carol	Female	30	04943	Flu

Table 1.1.: A fictional example of patient information held by a hospital.

the remaining data; e.g. there may be only a single person (Alice) who is female, 24 years old and lives at zip code 34290.

The second risk, which we focus on in this thesis, is leakage of private information due to correlation with public information. For example, some genders are more likely to get some diseases, some diseases are more frequent in certain age ranges, and infectious diseases may be more frequent in certain neighbourhoods. Although such correlation does not entirely give away what the private information (the diagnosed disease) was, it does allow an attacker to make a better guess about it. If an attacker can get their hands on enough "public" information, he may end up being able to make a very good guess at what the private information was.

To further prevent such correlation risk, the public information may be aggregated or distorted. For example, the data could be aggregated to only contain the first three digits of the zip code (342** instead of 34290) or a more general age range rather than a exact age (20–29 instead of 24).

The last example of aggregating ages is most likely not a very effective filter however: most diseases that correlate with age tend to correlate with age ranges, where it matters whether somebody is in their 20s or 60s, but does not matter much whether they are 21 or 26. In this case, we lost some useful information while doing little to prevent leakage of private information.

An effective filter should aggregate or distort the data in such a way that the utility of the data is mostly preserved while significantly reducing the amount of private information that is leaked. If the used filter isn't effective, you may end up significantly reducing the usefulness of your data while still leaking private information.

This raises the question of how to measure the effectiveness of a filter, and how to construct an effective filter. Unfortunately, to know how effective a filter is, you need to know the joint distribution between the public and private information. In the case of high-dimensional data, this distribution may be difficult to discover.

1.2. Generative adversarial networks

Machine learning is known to be useful for understanding high dimensional datasets. In particular, "Generative Adversarial Networks" [4] (GANs), were designed to be able to learn a complex distribution and then sample from them. They have often been applied to images, for example to learn the distribution of images of human faces, and then proceed to generate new images of human faces.

Generating images is a popular application of GANs. This is partially because it is an nice problem that generates sensational results, but also because GANs are relatively good at working with images. All machine learning methods need to trained on some data, but GANs in particular are very difficult to train [5][6], even with lots of data.

The GAN describes a framework that requires two neural networks (called the "generator" and "discriminator"); the architecture of those artificial neural networks is up to the user. GANs whose internal neural network have a deep convolutional architecture are called "Deep Convolutional Generative Adversarial Networks" [6] (DCGANs). Such

GANs have in practice turned out to be much easier to train than ordinary GANs whose internal networks do not have a convolutional architecture.

The disadvantage of DCGANs is that the dataset needs to have a structure suitable for convolutional networks. Images are a prime example of data that is suitable for convolutional networks, but many other kinds of data aren't.

A more recent invention known as the "Wasserstein GAN" [5] (WGAN) attempts to solve this by replacing the discriminator network with something else, called a "critic" network. Wasserstein GANs are considered to be easier to train than traditional GANs [7][8][9], and more importantly, they work with a far wider variety of neural network architectures, not just convolutional ones. Further research has discovered variants that improve Wasserstein GANs even further. The most notable one is the "Wasserstein GAN with Gradient Penalty" [10] (WGAN-GP), which is frequently considered to be better than the original Wasserstein GANs.

1.2.1. GANs for privacy

Although generative adversarial networks are meant to learn distributions, they unfortunately only learn distributions in the sense of "can sample new points from it"; they are unable to tell you much about the structure of the distribution such as "what is the likelihood of this sample being generated?". As such, even if a generative adversarial network learns the joint distribution, it is not of much use for classical methods of constructing privacy filters.

The article [1] proposes a method they call "generative adversarial privacy" wherein a variant of a generative adversarial network is used: rather than learning the probability distribution, their method directly tries to find an optimal privacy filter, and simultaneously includes a method to estimate how effective said filter is. Similar approaches can be found in [11] and [12]. Most of this thesis builds further on the work of [1]; we have attempted to reproduce their results, and explored a possible new variant of their techniques.

The method proposed by [1] is a variant of a traditional GAN that replaces the generator network with something different which they call a "privatiser". They also rename the discriminator network to "adversary", but it's function doesn't change. The privatiser is responsible for finding an optimal privacy filter, and the adversary is responsible for estimating the amount of information leaked in terms of the cross entropy [1], optimising the cross entropy is equivalent to optimising the amount of leaked mutual information.

In this thesis, we consider whether we can modify the approach taken by [1] to use a Wasserstein GAN as basis instead of a traditional GAN, in order to benefit from the usual advantages that a Wasserstein GAN has and significantly improve performance on datasets that are not suitable for convolutional networks.

The biggest problem is that a Wasserstein GAN would require the discriminator-based adversary with a critic-based adversary, and after doing so, it will measure the amount of leaked information in a unconventional metric based on the 1-Wasserstein distance rather than traditional units like cross entropy or mutual information.

This calls forth the question of whether we can be sure that our Wasserstein GAN variant is still performing properly. We investigated whether there are bounds on the amount of leaked information in terms of the 1-Wasserstein distance, and the general answer is no. However, we have proved that under certain additional conditions, it does become possible to bound the leaked information by the 1-Wasserstein distance, which is the main topic of our research.

1.3. Our contributions

In this thesis, we bring the following scientific contributions:

- We have tried to reproduce the results claimed by [1]. Although we conclude that the principle of their approach works, the results we reproduced are somewhat less spectacular than the results originally claimed;
- We introduce a new variant of [1]'s generative adversarial privacy networks which we call Wasserstein generative adversarial privacy networks;
- We prove theorems that relate the 1-Wasserstein distance between two distributions of certain classes to the amount of leaked information, giving theoretical justification for our proposed Wasserstein generative adversarial privacy networks;
- We give some counterexamples demonstrating that in general there is no direct relation between 1-Wasserstein distance and leaked information, justifying why our theorems put requirements on the classes of distributions to which they apply.

1.3.1. Structure of this thesis

In Chapter 2 we will talk about the generative adversarial privacy networks as introduced by [1] and attempt to reproduce their results. Then in Chapter 3 we will introduce Wasserstein GANs and use them to construct our own variant called Wasserstein genenerative adversarial privacy networks, and state theorems relating the performance of our variant to the amount of private information leaked in terms of mutual information, or more genenerally, f-information. Then in Chapter 4 we will talk about the practical relevance of our theorems and propose some avenues for future research.

In the appendices, we have put the details of our reproduction of [1]'s results, the proofs of the main theorems, and some counterexamples motivating the theorems.

2. Privacy networks

In this chapter, we will introduce generative adversarial networks [4] and how they can be adapted for privacy networks [1]. Thereafter, we will write about our outcome of attempting to reproduce [1]'s original results, where we conclude that the method works in principle, but our reproduced results are somewhat less spectacular than original result's impression.

2.1. Generative adversarial networks

Generative adversarial networks were introduced by [4] as a framework to generate samples from a complex distribution. As an example of a complex distribution, let us imagine $\mathbb{R}^{16\times 16}$ as the space of all 16×16 grayscale images, and a probability distribution H on $\mathbb{R}^{16\times 16}$ representing the distribution of 16×16 grayscale images of human faces one might encounter. The distribution H would assign a relatively high probability to images that represent normal human faces, low probability to images that represent uncommon faces (e.g. scarred ones), and zero probability to images that do not resemble human faces at all.

We assume that we do not know what the distribution H exactly looks like, but we are able to sample points from H, for example by scraping images off the internet or by taking photos of humans. We then want to try to fit the H into some distribution that can be sampled from with nothing but computational resources. A traditional statistical approach would be assuming H lies in parametrisable family of distributions, and then finding the parameters which are most likely to regenerate the samples. Traditional families of distributions like Gaussian distributions parametrised by their covariance matrices are clearly not complex enough to be able to realistically generate human faces.

2.1.1. Generators

In the generative neural network framework, a new parametrisable family of probability distributions is introduced: the family of all distributions that can be realised as the projection of Gaussian noise under a neural network.

For example, a "generator" network could be a neural network $G_\omega : \mathbb{R}^{100} \to \mathbb{R}^{16\times 16}$ that takes as input a vector of Gaussian noise (in this case a 100-dimensional vector), processes it through several layers with weights and biases determined by ω, and outputs a grayscale image in $\mathbb{R}^{16}$. If N is a random variable representing a Gaussian noise vector in $\mathbb{R}^{100}$, then $G_\omega(N)$ is a random variable on the space of grayscale images $\mathbb{R}^{16\times 16}$, which introduces a probability distribution $\mathrm{P}_{G_\omega(N)}$. The family of distributions is parametrised by ω.

For a given value of ω, we can sample from the distribution $\mathrm{P}_{G_\omega(N)}$ by first sampling Gaussian noise N and then computing $G_\omega(N)$. Using the maximum likelihood method, we should now look for the ω for which $\mathrm{P}_{G_\omega(N)}$ is the most likely distribution to generate our samples from H. Unfortunately, although we have a computationally efficient way to sample from $\mathrm{P}_{G_\omega(N)}$, we do not have a way to compute the likelihood of $\mathrm{P}_{G_\omega(N)}$ generating certain samples.

2.1.2. Discriminators

Instead of traditional methods like maximum likelihood estimation, the generative adversarial network framework introduces a new method: adding a second network called the discriminator. The discriminator is a network which takes as input a sample in $\mathbb{R}^{16\times 16}$ and outputs a guess on whether the sample was generated by H (real) or $G_\omega(N)$ (fake).

When the discriminator D_ψ is well trained, it becomes possible to judge the quality of the output of the generator by computing $D_\psi(G_\omega(N))$: if the output looks real according to the discriminator, then the generator is performing well; if the output looks fake according to the discriminator, the generator performs badly.

Using stochastic gradient descent, the parameters ω and ψ can be tuned: the parameter ψ is to be tuned to make the discriminator better at distinguishing real from fake examples, and the parameter ω is to be tuned to make the generator better fool the discriminator.

2.2. Privacy GANs

The preceding GAN scheme was intended as a method to generate new samples from a learned distribution. The goal of privacy networks is different: privacy networks try to create a new kind of distribution from an existing one. The architecture of privacy networks as introduced by [1] is pretty similar however.

We first repurpose the discriminator: instead of a network that tries to say whether a datapoint is real or fake, we use a network that tries to guess the private information from a datapoint of public information. Next, instead of generating new datapoints out of nothing, we make the generator modify real datapoints to make it difficult for the discriminator to guess the private information.

2.2.1. Notation

We will now establish the a notational framework, similar to the one used by [1]. We denote the public information that we want to share with the random variable $X \in \mathcal{X}$ where $\mathcal{X}$ is some metric space, and the private information that we don't want to share with $Y \in \mathcal{Y}$. We also define a random variable $N \in \mathcal{N}$ representing some random noise, for example N could be a standard Gaussian in $\mathbb{R}^k$.

We denote the privatiser with a function $G_\omega : \mathcal{X} \times \mathcal{Y} \times \mathcal{N} \to \mathcal{X}$ and the adversary with a function $D_\psi : \mathcal{X} \to \mathcal{Y}$.

The privatiser is a function that takes as input the public and private information X, Y, and gives a possibly random output. The noise N allows the privatiser to produce random output, which is important to prevent G_ω from being a deterministic and possibly reversible mapping. We use the random variable Z to denote the output of the privatiser: $Z = G_\omega(X, Y, N)$.

The discriminator, also called adversary, is supposed to take as input the output of the generator, and output its best guess of what Y was. We denote this guess with $\hat{Y} = D_\psi(Z) = D_\psi(G_\omega(X, Y, N))$. We further assume that there is a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, such that $\ell(Y, \hat{Y})$ determines how good the guess $\hat{Y}$ was. The goal of the adversary is to maximize the mutual information between $\hat{Y}$ and Y. The goal of the adversary is to minimize the loss $\ell(Y, \hat{Y})$.

Note that we did make some arbitrary choices in the spaces we defined. For example, we decided that the output of the privatiser has to lie in the same space as the public information, and that the adversary has to guess a value of the private information. This is not strictly necessary: it would be possible for the privatiser to output something in a different space $\mathcal{X}'$ as long as we have some means to estimate the distance between elements of $\mathcal{X}$ and $\mathcal{X}'$, and similarly the adversary could output something in a different space $\mathcal{Y}'$ as long as we can define a loss function ℓ between $\mathcal{Y}$ and $\mathcal{Y}'$.

However, such spaces $\mathcal{X}$ and $\mathcal{Y}$ would have to be chosen manually, as the privacy GAN method does not have a system to learn the best spaces $\mathcal{X}'$ and $\mathcal{Y}'$. As such, we do for simplicity assume that $\mathcal{X}' = \mathcal{X}$ and $\mathcal{Y}' = \mathcal{Y}$.

Although the above theoretical framework is quite general, in this thesis we tend to look at a more restricted subset. In particular, we usually assume that G_ω and D_ψ are neural networks and $\mathcal{X}$ and $\mathcal{Y}$ are Euclidean vector spaces.

2.2.2. Cross entropy

In machine learning classification tasks, a popular loss function is the cross-entropy loss. The cross-entropy loss is applicable on classification tasks with a finite amount of classes.

Assume that there are n classes and each entry corresponds to one of those classes. The classifier (the adversary in our case) should take as input a datapoint entry and guess to which class it belongs. The cross entropy loss expects the output of the classifier to be a set of n neurons, each with an activation in the interval $[0, 1]$ and the total activation summing up to 1. An example of an activation function which accomplishes this goal is the softmax activation function; assuming the output neurons have a raw input of x_i for neuron $i \in \{1, \ldots, n\}$, then the activation a_i of neuron i can be computed as:

$$a_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}}.$$

With the softmax activation function, the activation of each neuron i is usually interpreted as the probability that the input belongs to class i according to the classifier.

Assume the correct class is labelled using a one-hot vector $(q_i)_{i \in \{0,\ldots,n\}}$ such that $q_i = 1$ if the input belongs to class i and $q_i = 0$ otherwise, and the classifier activates the output neurons with activation $a_i \in [0,1]$ for $i = 1, \ldots, n$, then the cross-entropy loss can be computed as:

$$\ell(a, q) = -\sum_{i=1}^{n} q_i \ln a_i.$$

One good property of the cross-entropy loss is that when the cross-entropy loss is used for the adversary, the privatiser's goal of maximizing the adversary's loss is equivalent to minimizing the amount of mutual information between Y and Z [1].

2.2.3. Distance versus leaked information trade-off

In a classical GAN, the loss for the generator would be minus the loss of the discriminator: the worse the discriminator performs, the better the generator works and vice versa. If the adversary uses the cross-entropy loss, then the task of optimally fooling the adversary is equivalent to minimising the mutual information between Y and Z [1]. For a privatiser network, fooling the adversary is however not the only goal: it also needs to retain useful information in its output. After all, if there is no requirement for the privatiser to retain useful information, it may as well output the zero vector for any input, guaranteeing that the adversary won't be able to deduce anything.

This raises the question: how do we make sure useful information is retained? Ideally we'd have a function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that tells us how much useful information was retained or lost. The article [11] proposes to use another neural network for this purpose. However, in this thesis we are not going to focus on how such a function can be chosen; instead we assume that $\mathcal{X}$ is a metric space with metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and we want a bound on the average distance $\mathrm{E}[d(X, Z)]$ between the public information and its filtered version.

Given that we want to minimize two different variables, the leaked information and the distortion, there are several different optimization problems that may be constructed:

1. Minimize the leaked information constrained by an upper bound on $\mathrm{E}[d(X, Z)]$;
2. Minimize $\mathrm{E}[d(X, Z)]$ constrained by an upper bound on the leaked information;
3. Minimize some linear combination of the leaked information and $\mathrm{E}[d(X, Z)]$.

In order for the adversary's loss to be a viable estimate of the leaked information in terms of cross entropy, it is important that the adversary is well trained, which is requires having the privatiser constantly trying to maximise the adversary's loss. As such, option 2 may not be a good choice in this framework.

That leaves us with option 1 and 3. Option 3 is the easiest one to implement: it can be achieved by letting the privatiser's loss function be a linear combination of the adversary's loss and $d(X, Z)$. The disadvantage is that it requires you to decide how important average distance is compared to leaked information in terms of cross entropy; choosing a bad factor may lead to one of those two statistics getting neglected in favour of the other. Furthermore, leaked cross entropy may be a bit difficult to intuitively interpret,

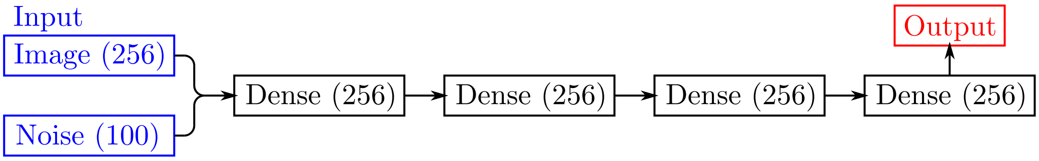


Figure 2.1.: A graph of the FFNP privatiser. The number in the parentheses is the amount of neurons in that layer.

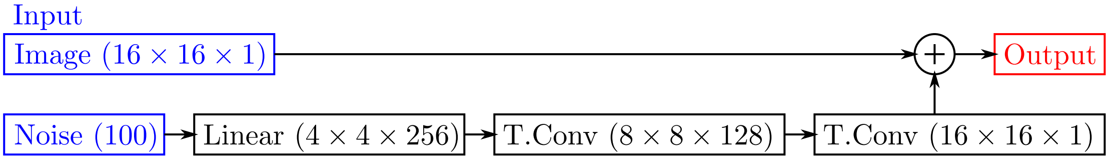


Figure 2.2.: A graph of the TCNNP privatiser. Processes Gaussian noise through a linear projection and two transposed convolutional layers.

even more so if we change the adversary's loss from "cross entropy" to "1-Wasserstein" distance as we will do later.

Option 1 is the one that was taken by [1]. It can be accomplished by adding a noise penalty to the privatiser whenever the distance between X and Z goes over a certain threshold:

$$\text{privatiser loss} = -\text{adversary loss} + \rho \cdot \max(0, d(X, Z) - \alpha).$$

In this formula, α and ρ are respectively a constant that decides how much distance between X and Z is acceptable, and a constant that decides how much loss is added when $d(X, Z)$ goes over that threshold. It may be desirable to increase the value of ρ as training goes on.

2.3. Reproducing the original results

We have attempted to reproduce the results from the original article [1]. The article describes two privatiser architecture used, which they call "FFNP" and "TCNNP", an adversary they used, a dataset used, and results they achieved.

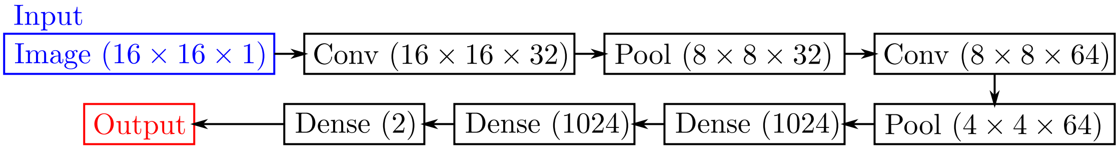


Figure 2.3.: A graph of the adversary. Processes a privatised image through convolutional, maxpool, and fully connected layers to guess whether the subject is male or female.

Privatiser	Distortion (target)	Distortion (real)	Adv. accuracy (test)	Adv. accuracy (training)
FFNP	0.004	0.0037	0.83	1.00
FFNP	0.008	0.0067	0.74	0.96
FFNP	0.012	0.011	0.60	0.86
FFNP	0.016	0.014	0.55	0.72
FFNP	0.020	0.016	0.55	0.64
TCNNP	0.004	0.00095	0.85	1.00
TCNNP	0.008	0.0071	0.81	0.99
TCNNP	0.012	0.0062	0.78	0.99
TCNNP	0.016	0.013	0.77	0.98
TCNNP	0.020	0.016	0.76	0.97

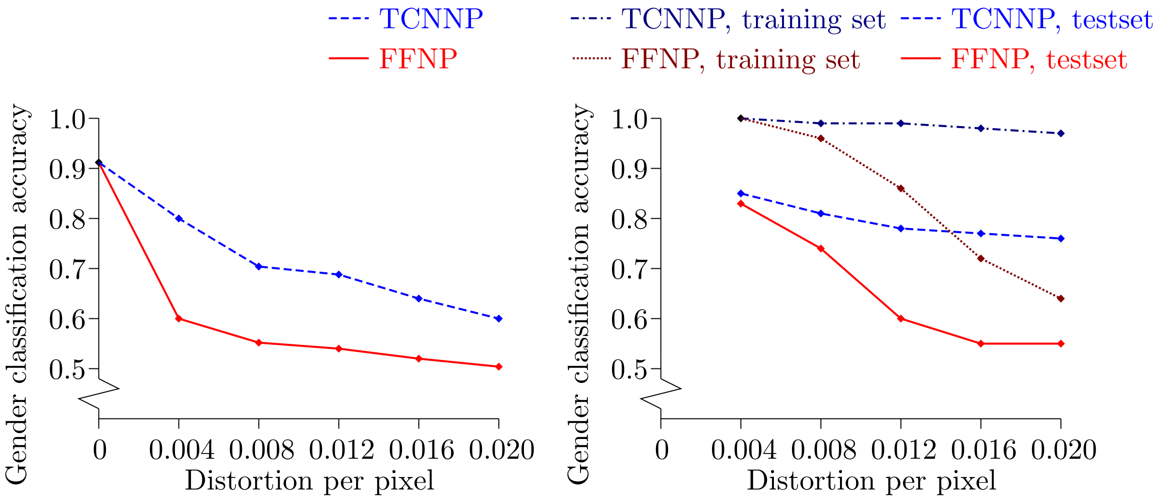
Table 2.1.: A table of the results of networks trained for 5,000 epochs with varying privatisers and allowed distortions. The allowed distortion is the mean square distortion per pixel the network is allowed to make on the training set before additional loss is assigned. The real distortion is the measured average amount of distortion the network added to the testset. The adversary accuracy indicates the probability the adversary correctly identifies the gender of a person, measured both on the training set and the testset.

The FFNP (Figure 2.1) is a network that takes the image and some random noise as input and processes them through four fully connected layers to give a privatized image as output. The TCNNP (Figure 2.2) is a network that takes random noise as input, processes it through transposed convolutional layers, and generates noise as output. The output noise of the TCNNP is then added to the image to form a privatised image. Both of them use the same adversary (Figure 2.3), which is a network that contains convolutional, maxpool and fully connected layers.

We tried reimplementing their networks as faithfully as possible, and training it on the same dataset. There were unfortunately a few unclear points in the architecture they described, for which we tried to make sensible assumptions. More about the assumptions we needed to make, along with our opinion about the network's design, can be read in Appendix A.

We have measured the adversary's performance on both the trainingset and the testset. The testset of 200 images is quite small, creating a significant amount of variance in the estimate of the adversary's accuracy. To get a better estimate, we have computed the adversary's accuracy by testing it on the testset for 1,000 epochs, which is sensible because the privatiser is stochastic: it will generate different outputs even when fed the same image multiple times. The results on the training set were computed as decaying averages of the adversary's performance while training.

The results have been written in Table 2.1. Things that we can immediately see is that the TCNNP privatiser is less effective than the FFNP privatiser, all networks are able to stay within their allowed distortion quota's, higher distortion quota's reduce the adversary's performance, and the adversary performs much better on the training set



(a) The results claimed by the original article [1].

(b) The results we managed to reproduce.

Figure 2.4.: A graph of the accuracy of the adversary compared to the target allowed distortion per pixel, using either a FFNP or a TCNNP privatiser. Our adversary's accuracy has been measured both on the testset and the training set, it is unclear on which the original results were measured.

than on the testset.

Although our results agree with the original article that the adversary's performance can become pretty low with enough allowed distortion, the curve in our experiments isn't as spectacular as the one claimed by the original article. First of all, we had to interpret "distortion per pixel" as "mean square distortion per pixel", which is not the most intuitive interpretation. Even after interpreting distortion that way, our adversary still performs better on both the training set and the testset than in the original article, which indicates that the privatiser may be less efficient than the original article indicates.

2.3.1. Distance Measure

The original article's results use an undefined term "distortion per pixel" to measure how different the original and filtered images are allowed to be. An intuitive interpretation may be: represent every image as a vector of grayscale brightness values between zero and one, then compute the average of the absolute difference in brightness values across all pixels.

This turns out to not be the article's authors' interpretation: their result claim that with a mere distortion of 0.008 per pixel, they can get the adversary's accuracy down to about 55%. If we imagine brightness as a value between 0–255, then a uniform distortion of 2 per pixel is barely visible to the human eye. If such a small distortion can completely throw off an adversary, then the adversary is probably performing poorly. In our experiments using this interpretation, the adversary would stay 83% accurate even with 0.020 distortion per pixel.

We then tried to reinterpret “distortion per pixel” as “mean square error”, or the average of the squared distance in brightness per pixel. When taking this interpretation, our results lined up closer to the original results claimed by the article.

2.3.2. Adversary performance

We have noticed that the adversary performs significantly better on the training dataset than on the testset. This means that the adversary is overfitting, which is very likely to happen with a small training set of 1740 entries.

Since the adversary may be significantly underperforming on the testset, we’d question the reliability of the figures we obtained on the testset. Unlike most other neural networks you train, in the adversary’s case it’s better to overestimate its performance than to underestimate it, so we may unorthodoxly want to measure the adversary’s performance on the training set instead of the testset.

However, even then there is an issue: the privatiser is most likely overperforming on the training set as well, and will perform significantly worse on the testset; we just don’t notice the difference between the performance on the training set and testset because the adversary performs even worse on the testset. We have measured “adversary with training set performance against privatiser with trainingset performance”, but it may be possible that “adversary with trainingset performance against privatiser with testset performance” performs even better.

This means that even the adversary’s performance statistics on the training set do not give an upper bound on how well the adversary might perform worst-case, or how much information we’re actually leaking. This calls into question the credibility of these figures.

3. Wasserstein generative adversarial privacy networks

In this chapter, we introduce our own variant of the generative adversarial privacy networks from the last chapter. We will first explain the Wasserstein GAN [5] and then use it to construct our own variant which we call Wasserstein generative adversarial privacy networks. We will then talk about the difficulties of comparing the performance of the Wasserstein generative adversarial privacy network to classical privacy metrics such as mutual information. We then introduce theorems that under certain circumstances do guarantee a relation between 1-Wasserstein distance and mutual-information or f-information. Finally, we talk about how to satisfy some of the theorem's requirements.

3.1. The 1-Wasserstein distance

The 1-Wasserstein distance is a metric between probability distributions. It is defined using the transport-theoretic notion of an optimal transport plan. Specifically, assume we have two random variables A and B on some shared metric space X with probability distributions respectively P_A and P_B, then the 1-Wasserstein distance $d_W(\mathrm{P}_A, \mathrm{P}_B)$ is defined as

$$d_W(\mathrm{P}_A, \mathrm{P}_B) = \inf_{(X,Y)\in\Pi(A,B)} \mathrm{E}\left[||X - Y||\right],$$

where $\Pi(A, B)$ is the set of all jointly distributed random variables whose marginal distributions are equal to those of A and B. Intuitively, this can be thought of as a transport problem where the probability mass of P_A needs to be optimally transported to the probability mass of P_B, and the cost of transporting mass is the amount of mass to be transported times the distance it must be transported over. The 1-Wasserstein distance between two random variables is the cost of the optimal transport plan.

The 1-Wasserstein distance can alternatively be computed using the Kantorovich-Rubinstein duality [13], which states that the 1-Wasserstein distance between two probability distributions on a compact space X is equal to

$$d_W(\mathrm{P}_A, \mathrm{P}_B) = \sup_{\substack{f:X\to\mathbb{R} \\ f \text{ 1-Lipschitz continuous}}} \mathrm{E}_{x\sim\mathrm{P}_A}[f(x)] - \mathrm{E}_{x\sim\mathrm{P}_B}[f(x)].$$

3.2. The Wasserstein GAN

Remember how generative adversarial networks were introduced as a method to learn a distribution $G_\omega(N)$ that is similar to an unknown distribution H from which we have samples. The classical approach is to create a discriminator D_ψ which guesses for each sample how likely it was to have been generated by $G_\omega(N)$ or H.

The Wasserstein GAN is a more recent variant of the classical GAN introduced by [5]. In a Wasserstein GAN, the goal of the discriminator is no longer to find the likelihood that a single sample was generated by one distribution or another, but rather to estimate the "distance" between the two distributions. In particular, the 1-Wasserstein distance, also known as the earth mover distance, is used.

When using the Kantorovich-Rubenstein duality to compute the 1-Wasserstein distance as the supremum over 1-Lipschitz-continuous functions f, the optimal function f can be approximated using a neural network; this is the main idea behind the Wasserstein GAN: we require the discriminator (now renamed "critic") D_ψ to be a Lipschitz-continuous function, and then compute the loss as

$$\begin{aligned}\text{critic loss} &= -E[D_\psi(H) - D_\psi(G_\omega(N))],\\ \text{generator loss} &= E[D_\psi(H) - D_\psi(G_\omega(N))].\end{aligned}$$

When the critic works optimally, the loss should be equal to a factor of the 1-Wasserstein distance between H (real) and $G_\omega(N)$ (fake), said factor depending on the Lipschitz constant of D_ψ. The critic is trained to become better at estimating the 1-Wasserstein between the real and fake samples, and the generator is trained to minimise the 1-Wasserstein distance between real and fake samples. The way to enforce D_ψ to be Lipschitz continuous varies between the original version of the Wasserstein GAN [5] and the derivatives like the WGAN-GP [10], and can include techniques such as constraining the weights of the neural networks or adding a loss penalty if the Lipschitz constraint is violated.

Besides being feasible to compute thanks to the Kantorovich-Rubenstein duality, the 1-Wasserstein metric is useful because it can give a meaningful distance between any two distributions. Other conventional distance measures distance measures such as the Kullback-Leibler divergence [14] or total variation distance will quickly assign an complete dissimilarity between P_A and P_B when the support of those distributions is disjoint. The 1-Wasserstein distance on the other hand may still assign a low distance between distributions with disjoint supports provided those supports lie close to each other. When used in combination with stochastic gradient descent, the 1-Wasserstein metric d_W will tell us in which direction the supports need to move to further reduce their distance.

Importantly, when the critic works well but the generator doesn't, the critic can still backpropagate sensible gradients, unlike a traditional discriminator which might take values of approximately "0" and "1" at the entire supports of $G_\omega(N)$ and H. This means that there is no problem with overtraining an critic, removing one of the big causes of training instability in traditional GANs.

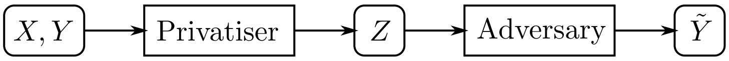


Figure 3.1.: A computation scheme of how an ordinary privatiser GAN as used by [1].

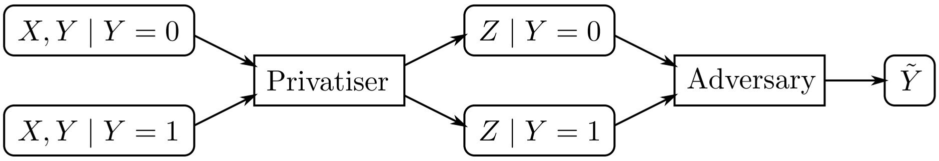


Figure 3.2.: The distribution of the output of the decomposed into two seperate distributions, seperated on basis of what the underlying private variable was.

3.3. The Wasserstein generative adversarial privacy networks

In a classical Wasserstein GAN, the loss of the adversary approximates the 1-Wasserstein distance between the distribution of real samples and the distribution of fake samples. In the context of privacy networks, there are no “real” or “fake” samples, as all samples are distorted versions of real information.

However, when the private information is binary, for example trying to distinguish between pictures of male human faces and female human faces, the adversary’s task still reduces to trying to trying to differentiate between two different distributions, in this particular case the distribution of all privatised images of male faces and the distribution of all privatised images of female faces.

Let us use some formal notation now. Let the random variable Y represent the private variable and be Bernouilli distributed, so either $Y = 0$ or $Y = 1$. Entries of public information X sampled from the dataset follow a certain distribution P_X. We can split the dataset into two subsets: one subset containing all entries where $Y = 0$ and another subset containing all entries where $Y = 1$. The joint distribution of public information is a combination of the distributions of the subsets:

$$\mathrm{P}_X = \mathrm{P}(Y = 0) \cdot \mathrm{P}_{X|Y=0} + \mathrm{P}(Y = 1) \cdot \mathrm{P}_{X|Y=1}.$$

Let the random variable Z be the output of the privatiser. Likewise, Z follows a distribution P_Z, which can be decomposed into the distribution of the output of the privatiser when given an input with $Y = 0$ and the distribution of the output of the privatiser when given an input with $Y = 1$:

$$\mathrm{P}_Z = \mathrm{P}(Y = 0) \cdot \mathrm{P}_{Z|Y=0} + \mathrm{P}(Y = 1) \cdot \mathrm{P}_{Z|Y=1}.$$

In the original approach by [1], the adversary would now get as input a sample of Z and be tasked with guessing $\tilde{Y}$. Assuming the adversary works optimally, information about the mutual information between X and Y could be estimated on basis of how well the adversary was able to guess Y. In our approach, instead of asking the adversary to estimate Y, we ask it to estimate the 1-Wasserstein distance between the two distributions $\mathrm{P}_{Z|Y=0}$ and $\mathrm{P}_{Z|Y=1}$.

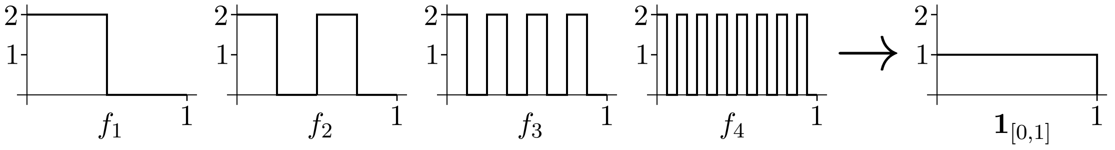


Figure 3.3.: The graphs of the a series of probability density functions whose associated random variables converge to the uniform distribution on $[0, 1]$.

The general idea is that if we can train the privatiser to generate output distributions such that $\mathrm{P}_{Z|Y=0} = \mathrm{P}_{Z|Y=1}$, then Z would be independent of Y and we would achieve perfect privacy. However, in practice it we won't be able to achieve those constraints using artificial neural networks, so instead we have to settle for "the distance between $\mathrm{P}_{Z|Y=0}$ and $\mathrm{P}_{Z|Y=0}$ is very small." We will prove that under sufficient conditions, the leaked information about Y converges to zero if the 1-Wasserstein distance between $\mathrm{P}_{Z|Y=0}$ and $\mathrm{P}_{Z|Y=1}$ converges to zero.

3.4. Wasserstein distance versus leaked information

This approach does call forth the question: $Z \mid Y = 1$ is very close to $Z \mid Y = 0$ in terms of the 1-Wasserstein distance, does that give us any guarantees on how difficult it is to estimate the private information given samples of privatised information? As we noted, the 1-Wasserstein distance can assign small distances between distributions with disjoint supports. This may be useful for training networks, but may on the other hand prevent it from being an useful privacy metric.

Unfortunately, it turns out there is no direct connection between the Wasserstein distance between two distributions and the amount of information leaked. We will give a counterexample with a series of random variables whose 1-Wasserstein distance becomes arbitrarily small, but nevertheless leak constant amounts of information, and then investigate what further assumptions we need to make to get guarantees on the leaked information in terms of 1-Wasserstein distance.

3.4.1. Counterexample

Let $(Z_n \mid Y = 1)_{n\in\mathbb{N}}$ be uniform on the interval $[0, 1]$ for all $n \in \mathbb{N}$ and let the random variables $(Z_n \mid Y = 0)_{n\in\mathbb{N}}$ have the density functions

$$f_n(x) = \begin{cases} 2 & \text{if } x \in [0,1] \text{ and } \lfloor 2^n x \rfloor \equiv 0 \mod 2, \\ 0 & \text{otherwise.} \end{cases}$$

where $\lfloor\cdot\rfloor$ denotes the floor function.

The first four functions of this series have been drawn in Figure 3.3. The random variables $Z_n \mid Y = 0$ converge in 1-Wasserstein distance to the uniform distribution on $[0, 1]$. To see this, remember that the 1-Wasserstein distance between two random variables can be visualised as the cost of the optimal transport plan that turns the probability mass of one random variable into the other.

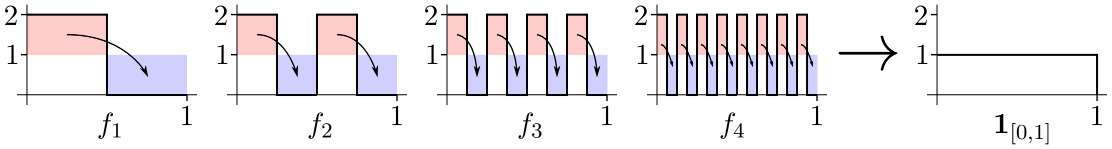


Figure 3.4.: To turn F_n into the uniform distribution, the red probability mass surplus mass must be transported to the blue probability mass deficit. The "cost" of this transport plan is the 1-Wasserstein distance.

One example of a transport plan that turns $Z_n \mid Y = 0$ into a uniform distribution has been drawn in figure 3.4. Note that exactly half of the probability mass is already in the right spot and doesn't need to be moved, whereas the other half of the probability mass needs to be transported over distance 2^{-n}. The amount of mass that needs to be transported stays constant, but the distance it needs to be transported over approaches zero as $n \to \infty$.

As such, the cost of the transport plan approaches zero as $n \to \infty$ and the 1-Wasserstein distance between $Z_n \mid Y = 0$ and the uniform distribution approach zero, which happens to be $Z_n \mid Y = 1$. Hence the 1-Wasserstein distance between the random variables $Z_n \mid Y = 0$ and $Z_n \mid Y = 0$ converges to zero as $n \to \infty$.

It is however obvious that the density functions f_n do not converge pointwisely or uniformly to $\mathbf{1}_{[0,1]}$ at all, and moreover for any given n it is easy to see that information leaks: suppose a sample of Z_n lies outside the support of $Z_n \mid Y = 0$ (which happens with probability $\frac{1}{4}$ if Y is Bernouilli($\frac{1}{2}$)-distributed) then we are guaranteed that $Y = 1$, whereas if the sample of Z_n lies in the support of $Z_n \mid Y = 0$ then there is a $\frac{2}{3}$ chance that $Y = 0$.

We see that merely having a tiny 1-Wasserstein distance between $Z \mid Y = 0$ and $Z \mid Y = 1$ is not sufficient to guarantee that Z leaks no information about Y.

3.4.2. Gaussian noise

In the previous example, we managed to get the 1-Wasserstein distance arbitrarily small by rapidly alternating source and sink areas to reduce the distance over which mass had to be moved without reducing the density of the mass.

Inspired by how [1] proposed to achieve privacy by adding Gaussian noise to the output, we noticed that an counterexample like Figure 3.3 would be infeasible if we required a constant amount of Gaussian noise to be added to all Z_n: it would smooth the probability density over a certain area making rapid alternations in probability density impossible.

You may wonder whether adding Gaussian noise to Z is enough to guarantee some continuity of the leaked information in terms of the 1-Wasserstein distance. This is indeed still an open question.

Adding Gaussian noise to a random variable does guarantee that the result is continuously distributed with a Lipschitz-continuous probability density function (proof in Section 3.7). If the output of the privatiser's neural network is Z', then we can define $Z = Z' + n$ and consider Z to be the actual output of the privatiser. Assuming that the output of the privatiser is continuously distributed with a Lipschitz continuous probability density function, then we can, under a few more conditions, achieve bounds on the amount of leaked information in terms of the 1-Wasserstein distance, as we will prove in Section 3.6.

3.5. Intermezzo: f-information

So far, we've been talking about leaked information in terms of mutual information, but there is a more general kind of information called f-information, which we will use in the formulations of the main theorems.

Remember that mutual information between random variables P and Q is is equal to the KL-divergence between the jointly distributed variable (P, Q) and an independently distributed random variable (P^*, Q^*) where the marginal distributions of P^* and Q^* are equal to those of P and Q. The KL-divergence between two random variables P and Q can be computed as

$$D_{\mathrm{KL}}(P||Q) = \int \ln\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right)\, \mathrm{d}P,$$

where $\mathrm{d}P/\mathrm{d}Q$ refers to the Radon-Nikodym derivative of P with respect to Q. For a convex function $f : (0, \infty) \to \mathbb{R}$ with $f(1) = 0$, the f-divergence is defined as

$$D_f(P||Q) = \int f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right)\, \mathrm{d}Q.$$

If both P and Q are absolutely continuous with respect to some σ-finite measure μ, then this can alternatively be computed as

$$D_f(P||Q) = \int \frac{\mathrm{d}Q}{\mathrm{d}\mu} f\left(\frac{\mathrm{d}P/\mathrm{d}\mu}{\mathrm{d}Q/\mathrm{d}\mu}\right) \mathrm{d}\mu,$$

where the integrand is appropriately specified at the points where the densities $\mathrm{d}P/\mathrm{d}\mu$ and/or $\mathrm{d}Q/\mathrm{d}\mu$ are zero [15].

In particular, for the function $f(t) = t \ln t$, this is equivalent to the KL-divergence [15]. The f-information is to f-divergence as mutual information is to KL-divergence: the f-information between P and Q is defined as the f-divergence between the joint distribution (P, Q) and the product of their marginal distributions. In particular, for $f(t) = t \ln t$, the f-information is equal to the mutual information.

3.6. Main results

We have managed to prove absolute continuity of the leaked information with respect to the 1-Wasserstein distance. In particular, if we have a series of random variables $(Z_n)_{n\in\mathbb{N}}$ which satisfy the following properties:

- The distributions of $Z_n \mid Y = 0$ and $Z_n \mid Y = 1$ are continuous, and their probability density functions are Lipschitz continuous;
- The probability measures of the distributions of $Z_n \mid Y = 0$ and $Z_n \mid Y = 1$ are tight;
- The 1-Wasserstein distance between the distributions of $Z_n \mid Y = 0$ and $Z_n \mid Y = 1$ converges to zero as $n \to \infty$.

Then the leaked information, i.e. the mutual information between Z_n and Y, converges to zero as $n \to \infty$. In fact, we've managed to prove a slightly stronger claim: the f-information between Z_n and Y converges to zero as $n \to \infty$.

Formally, we've stated our theorems in two steps:

Theorem 1. *Let $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ be series of random variables on $\mathbb{R}^k$ with continuous probability distributions, whose probability density functions $a_n, b_n : \mathbb{R}^k \to \mathbb{R}$ are L-Lipschitz continuous. If A_n converges to B_n under the 1-Wasserstein metric d_W, i.e. $\lim_{n\to\infty} d_W(A_n, B_n) = 0$, then the probability densities a_n converge uniformly to b_n as $n \to \infty$, i.e. $\lim_{n\to\infty} ||a_n - b_n||_\infty = \lim_{n\to\infty} \sup_{x\in\mathbb{R}^k} ||a_n(x) - b_n(x)|| = 0$.*

Theorem 2. *Let $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ be continuously distributed random variables with continuous probability density functions a_n and b_n such that a_n converges uniformly to b_n as $n \to \infty$. Let $V \sim \mathrm{Bernoulli}(\frac{1}{2})$. Assume further that for at least one of the series $(A_n)_{n\in\mathbb{N}}$, $(B_n)_{n\in\mathbb{N}}$, the probability distributions of said series are tight; i.e. in the case of $(A_n)_{n\in\mathbb{N}}$ it means for all $\varepsilon > 0$ there exists a compact set $K_\varepsilon \subset \mathbb{R}^k$ such that for all $n \in \mathbb{N}$ we have $\mathrm{P}(A_n \in K_\varepsilon) > 1 - \varepsilon$.*

Let $f : (0, \infty) \to \mathbb{R}$ be a convex function such that $f(1) = 0$ and $\lim_{x\to 0} f(x) < \infty$. Let $\mathcal{I}_n^f$ be the f-information between the random variables $VA_n + (1 - V)B_n$ and V. Then $\lim_{n\to\infty} \mathcal{I}_n^f = 0$.

The proofs of these theorems can be found in Appendix B.

The output of a privatiser which has been trained for n steps can be seen as a random variable Z_n, and A_n and B_n can be seen as the random variables $Z_n \mid Y = 1$ and $Z_n \mid Y = 0$ respectively. The above theorem tells us that if the distributions of A_n and B_n have certain properties (continuously distributed with Lipschitz continuous density functions) and their support doesn't get too large (the series must be tight), then convergence under the 1-Wasserstein distance implies that the leaked information goes to zero.

The above theorem however does not talk about the rate of convergence. We hypothesize that the rate of convergence is at least $O(-\sqrt{x} \ln x)$, and have mostly proven this except we still rely on one unproven assumption (see Appendix C). We furthermore assume that the supports of A_n and B_n are have finite and uniformly bounded measure, rather than just tight support.

Proposition 3. *Assume that Assumption C.1 is true. Let $f : (0, \infty) \to \mathbb{R}$ be a convex function such that $\lim_{x\to 0^+} f(x) < \infty$. Let A_n and B_n be continuously distributed random variables on $\mathbb{R}^k$ with probability density functions a_n and b_n such that the following holds:*

- *There is a constant $L \in \mathbb{R}$ such that a_n and b_n are L-Lipschitz continuous;*
- *There exists a set $K \subset \mathbb{R}^k$ with $\lambda^k(K) < \infty$ such that* $\operatorname{supp} a_n \subset K$ *and* $\operatorname{supp} b_n \subset K$*;*
- *The 1-Wasserstein distance $d_W(A_n, B_n)$ is sufficiently small, in particular, such that*
$$d_W(A_n, B_n) \leq \frac{1}{8L^2\lambda^k(K)}.$$

Let $Z \sim$ Bernouilli$(\frac{1}{2})$. Then there exists constants c_1, c_2 whose value depend only on f, $\lambda^k(K)$ and L, such that the f-information $\mathcal{I}_n^f$ between Z and $ZA_n + (1-Z)B_n$ is upper bounded by

$$\mathcal{I}_n^f \leq c_1 \cdot \sqrt{d_W(A_n, B_n)}\big(c_2 - \ln d_W(A_n, B_n)\big).$$

Corollary 4. *In particular, if $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ are series of random variables such that for all $n \in \mathbb{N}$, A_n and B_n satisfy the requirements of Proposition 3 with uniform values for L and $\lambda^k(K)$, then*

$$\mathcal{I}_n^f = O\big(-\sqrt{d_W(A_n, B_n)} \ln d_W(A_n, B_n)\big).$$

The proof of this theorem can be found in Appendix C.

These theorems are more of theoretical relevance than practical relevance. The convergence rate $-\sqrt{x}\ln x$ is not very fast and the constant c_1 in the above theorem is huge in most practical situations. With the limited numerical accuracy a computer has, it is unlikely that the term $c_1 \cdot \sqrt{d_W(A_n, B_n)}\big(c_2 - \ln d_W(A_n, B_n)\big)$ will ever reach a value smaller than 1 no matter how long you train a network.

3.7. Lipschitz continuous probability density

All the stated theorems work with continuous distributions $Z \mid Y = 0$ and $Z \mid Y = 1$ which have Lipschitz-continuous probability density functions. This is the biggest assumption we have to make, and a particularly problematic one as the output of neural networks will, in general, not be continuously distributed.

We can turn any distribution into a continuous distribution with Lipschitz continuous probability density by adding Gaussian noise to it, and the Lipschitz constant of the probability density will be upper-bounded by that of the noise added. See Theorem 5.

Theorem 5. *Let X be a random variable on $\mathbb{R}^k$ and let N be a Gaussian random variable on $\mathbb{R}^k$ whose probability density is L-Lipschitz continuous. Then the random variable $X + N$ is continuously distributed and its density function is L-Lipschitz continuous.*

This theorem can be used to make the output of the privatiser continuously distributed by adding Gaussian noise N: if Z' is the direct output of a privatiser's neural network, then we can define $Z = Z' + N$ and consider Z to be the privatiser's real output which is to be fed into the adversary. This guarantees that Z is continuously distributed and the Lipschitz constant of its probability density function is bounded, no matter what distribution Z' has.

Although Theorem 5 is not a big result and most likely already known, we will provide proof for it anyway. First we introduce some lemmas:

Lemma 3.1. *For any continuous distributed random variable N, the random variable $X + N$ is continuously distributed.*

Proof. We will first show that the probability density of $X + N$ is absolutely continuous with respect to the k-dimensional Lebesgue measure λ^k. Let $A \subset \mathbb{R}^k$ such that $\lambda^k(A) = 0$. Then

$$\begin{aligned} \mathrm{P}(X + N \in A) &= \iint \mathbf{1}_A(x + t) \ \mathrm{dP}_N(t) \ \mathrm{dP}_X(X) \\ &= \int \mathrm{P}_N(A - x) \ \mathrm{dP}_X(x), \end{aligned}$$

Where $A - x = \{a - x \mid a \in A\}$. Since the Lebesgue measure is invariant under translation, we have $\lambda^k(A + x) = 0$, and since N is continuously distributed, $\mathrm{P}_N(A - x) = 0$, thus $P(X + N \in A) = 0$. Since the distribution of $X + N$ is a probability measure absolutely continuous with respect to λ^k, it is continuously distributed due to the Radon-Nikodym theorem. □

Lemma 3.2. *If Y is a continuous random variable on $\mathbb{R}^k$ and N is an independent continuous random variable on $\mathbb{R}^k$ whose probability density function f is L-Lipschitz continuous, then the probability density function of $Y + N$ is L-Lipschitz continuous.*

Proof. Since Y and N are independent continuous random variables with probability densities respectively g and f, the probability density h of their sum $Y + N$ can be computed using a convolution:

$$h(x) = \int g(t) f(x - t) \ \mathrm{d}t.$$

We will now prove that h is L-Lipschitz continuous. Let $x, y \in \mathbb{R}^k$:

$$\begin{aligned} ||h(x) - h(y)|| &= \left|\left| \int g(t) f(x - t) \ \mathrm{d}t - \int g(t) f(y - t) \ \mathrm{d}t \right|\right| \\ &= \left|\left| \int g(t) \big(f(x - t) - f(y - t) \big) \ \mathrm{d}t \right|\right| \\ &\leq \int \left|\left| g(t) \big(f(x - t) - f(y - t) \big) \right|\right| \ \mathrm{d}t \\ &= \int g(t) \cdot ||f(x - t) - f(y - t)|| \ \mathrm{d}t \\ &\leq \int g(t) \cdot L \cdot ||x - y|| \ \mathrm{d}t \\ &= L \cdot ||x - y|| \,. \end{aligned}$$

□

Proof of Theorem 5. A Gaussian random variable N can be decomposed into independent Gaussian random variables N_1 and N_2 such that $N = N_1 + N_2$ and the Lipschitz constant of N_2 comes arbitrarily close to L. Then $X + N = (X + N_1) + N_2$ where $X + N_1$ is continously distributed according to the first lemma, and $(X + N_1) + N_2$ has a Lipschitz constant no bigger than N_2 due to the second lemma. Since the Lipschitz constant of N_2 can be made arbitrarily close to that of N, it follows that the Lipschitz constant of $(X + N_1) + N_2$ is no bigger than something arbitrarily close to that of N, and hence the Lipschitz constant of $X + N = (X + N_1) + N_2$ is no bigger than that of N. □

4. Discussion

We have investigated whether the 1-Wasserstein distance can be used instead of the cross-entropy loss when measuring the amount of information leaked by privacy networks, and have for certain classes of distributions demonstrated a theoretical guarantee that the leaked information will go to zero if the 1-Wasserstein distance goes to zero. Additionally, under a certain assumption, we have also managed to prove a rate of convergence for certain classes of distributions. In this chapter, we will discuss the practical relevance of those results and propose a couple of points for future research.

4.1. Theory versus practice

These results are mostly of theoretical importance, as they only apply when the 1-Wasserstein distance gets really small; smaller than is likely possible to be achieved in practice. Nevertheless, knowing that there is a theoretical guarantee that the leaked information is absolutely continuous with respect to the 1-Wasserstein distance, there is hope that in practice leaked information converges significantly faster to zero than in the theoretical worst case scenario.

We are lacking practical research into how a Wasserstein privacy GAN compares to a traditional privacy GAN. For future research, we propose the following experiment to check whether a privatiser that gives a small 1-Wasserstein distance also leaks little information:

1. First, train a privatiser with a Wasserstein adversary to convergence;
2. Then replace the Wasserstein adversary with a cross-entropy adversary, fix the privatiser and train only the adversary to convergence;
3. Estimate the amount of information the privatiser leaked based on the performance of the cross-entropy adversary.

4.2. Lipschitz continuity versus Gaussian noise

All of our theorems apply to continuous distributions which have Lipschitz-continuous probability density functions. One way to turn any probability distribution into a continuous one with a Lipschitz continuous probability density function is by adding Gaussian noise. However, the class of “distributions that contain Gaussian noise” is a strict subset of the class of “continuous distributions with Lipschitz-continuous probability density functions”.

This makes us wonder whether better results are possible when working with the class of distributions that contain Gaussian noise. For example, it may be possible that when Gaussian noise is assumed, the rate of convergence increases, or that other assumptions such as tightness are no longer needed. This may be a subject of future research.

4.3. Extension to multiple classes of private information

Our proposed Wasserstein generative adversarial privacy networks only work when the private information Y is binary, because we can compute the 1-Wasserstein distance only between two distributions $Z \mid Y = 0$ and $Z \mid Y = 1$ simultaneously.

Now suppose Y was discrete and took values in $\{0, 1, 2, 3\}$. Then there are four distributions $Z \mid Y = 0$, $Z \mid Y = 1$, $Z \mid Y = 2$, and $Z \mid Y = 3$ that we want to be indistinguishable from each other. It might be possible to now use six adversaries, each adversary estimating the distance between one pair of marginal distributions, and then somehow manage to optimize them simultaneously. However, even if this is possible, it doesn't scale very well: given n classes, the amount of adversaries needed is $\frac{1}{2}n^2 - \frac{1}{2}n$.

We noticed that our approach is somewhat similar to the t-closeness principle. The concept of t-closeness was introduced by [16]; their goal is to make the distribution of the private information look similar for each entry of privatised data, whereas we are trying to make the distribution of privatised data look similar for each class of private data. Symbolically, they make the distribution of $Y \mid Z = z$ be close to the distribution of Y for all observable values of z. This inspires us to the idea: instead of making the distribution of $Z \mid Y = 0$ close to the distribution of $Z \mid Y = 1$, we could try making the distribution of $Z \mid Y = y$ close to the distribution of Z for all private classes y.

In this case, we would need only four adversaries: one that minimizes the 1-Wasserstein distance $d_W(Z \mid Y = 0, Z)$, another one that minimizes the distance $d_W(Z \mid Y = 1, Z)$, and so on. In this case, we need only n adversaries for n classes of private data, which scales better.

Of course this does not solve the question of whether it is possible to effectively train multiple adversaries simultaneously, which could be an avenue for future research.

4.4. Optimality of neural networks

So far, we have been relying on the assumption that the adversary works optimally in order to achieve good privacy and be able to estimate how much information leaks. This induces the question whether such an assumption is justified: if the adversary does not work optimally, it is possible to severely underestimate the amount of information that gets leaked.

There are two issues that can prevent the adversary from working well. First, there is the question whether any configuration of an adversary network's weights is able to express the theoretically optimal adversary function. The leaked information can be computed as $\min_D \ell(D(Z), Y)$ and any adversary we train belongs to a family D_ψ parametrised by ψ. Are we sure there exists a ψ such that $D_\psi = \text{argmin}_D \ell(D(Z), Y)$?

The universal approximation theorem [17] roughly states that a sufficiently big neural network with at least one hidden layer is able to approximate all continuous functions on a compact domain with arbitrary precision. Because of this, we hope that with a large enough neural network, D_ψ is able to approximate the theoretically optimal adversary with arbitrary precision. This assumes that the optimal adversary is a continuous function, an assumption which we haven't verified.

The second question is: even if such a ψ exists, does training the adversary cause it to converge to those weights? The usual training algorithms for neural networks are intended to make it converge to a local optimum, which is not necessarily a global optimum. Hence, it is possible that a fully trained adversary performs significantly worse than it theoretically could.

Fortunately there is a hypothesis that for large neural networks, most local optimums are approximately equally good. This hypothesis is unproven, but is supported by [18], which proves this claim for a model which somewhat resembles neural networks. This gives hope that a sufficiently big well-trained adversary will perform somewhat close to optimally.

Nevertheless, in the end there is no guarantee that the adversary is working well; this is a major issue with our approach for which we don't have a solution.

A. Reproduction

We have attempted to reproduce the results from [1]. We have written about our outcome in Section 2.3. There were some unclear details regarding their methodology where we had to make some sensible assumptions. This appendix exists to clarify those assumptions. From our own practical intuition, we do not think their networks were well designed. At the end of this appendix, we also give some criticism about the networks' design.

A.1. The dataset

The article claims that the dataset is the GENKI dataset which contains 1,940 grayscale images of resolution 16 × 16, and their goal of the privatiser is to hide the gender of the person on the image. The problem is that the MPLab GENKI database [19] contains about 4,000–7,500 images (depending on which subset you take), the images are in colour, have a resolution of approximately 180 × 192 (differs per image), and the dataset contains no labels for the gender of the subject. In short, their description of the dataset does not seem to match the official GENKI dataset.

We have however managed to find a dataset that matches their description at an unrelated Github repository [20]. We presume that this is the same dataset they used.

A.2. The privatiser

The article [1] proposes two different privatiser networks which they call FFNP and TCNNP, both of them would be followed up by the same adversary network. We have implemented both of them.

- Network 1: input → FFNP → Adversary;
- Network 2: input → TCNNP → Adversary.

There are some ambiguities in the description of the original article, so here we describe their network along with the assumptions we made in greater detail.

A.2.1. First privatiser: FFNP

The first privatiser, drawn in Figure A.1 they propose is what they call the FFNP model (feedforward neural network privatiser). It is a fully connected feed-forward network with three hidden layers and batch normalisation.

The input image is a 16 × 16 × 1 tensor, representing a height of 16, width of 16 and single channel (grayscale) of information per pixel. It is then reshaped into a vector of

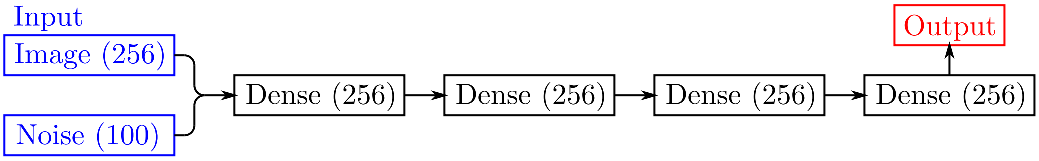


Figure A.1.: A graph of the FFNP privatiser. The number in the parentheses is the amount of neurons in that layer.

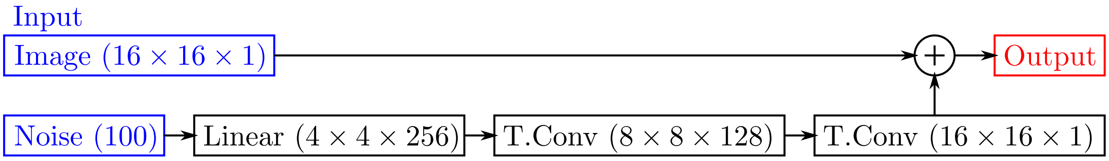


Figure A.2.: A graph of the TCNNP privatiser. Processes Gaussian noise through a linear projection and two transposed convolutional layers.

length 256, and has 100 standard normally distributed random variables concatenated, resulting in a vector of length 356. This vector becomes the input of the network.

This input vector is then fed into four fully connected layers, each layer contains 256 neurons, uses Leaky ReLU activation and uses batch normalisation. The original article does not tell us what the slope of the Leaky ReLU is for inputs below zero, so we assume the Tensorflow default of 0.2:

$$\text{LReLU}(x) = \begin{cases} x & \text{for } x \geq 0, \\ \frac{1}{5}x & \text{for } x < 0. \end{cases}$$

The first three layers also employ batch normalisation as introduced in [21]. The goal of batch normalisation is to normalise the activation of each neuron such that in a single batch, each neuron i always has mean activation μ_i and standard deviation γ_i, where μ_i and γ_i are trainable parameters. Note that this normalisation occurs before the activation function is applied.

$$\text{layer}_2 = \text{LReLU}(\text{BatchNorm}(A_2 \cdot \text{layer}_1)).$$

The output of the fourth layer, which is a vector of length 256, is subsequently reshaped into an image of size $16 \times 16 \times 1$, which is the output of the network. This fourth layer still uses Leaky ReLU activation, though it doesn't use batch normalisation. Of course Leaky ReLU is a terrible activation function for the output layer as the output should lie in the range $[0, 1]$ whereas the Leaky ReLU has a range of $(-\infty, \infty)$, but it is what the original article uses.

A.2.2. Second privatiser: TCNNP

The second privatiser they propose is the TCNNP (transposed convolutional neural network privatiser). This network is focussed on generating noise patterns with a complex distribution rather than on processing the input image.

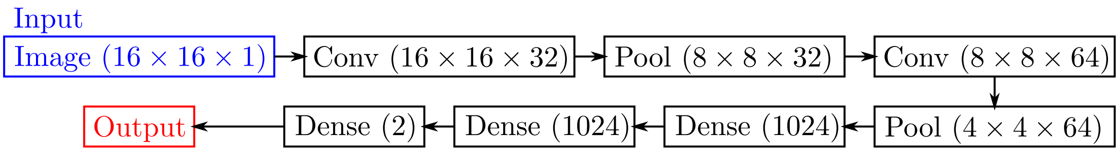


Figure A.3.: A graph of the adversary. Processes a privatised image through convolutional, maxpool, and fully connected layers to guess whether the subject is male or female.

The input image is again a $16 \times 16 \times 1$ tensor, but it is not fed into the network. Instead, the input layer of this network is merely a noise vector of 100 normally distributed random variables.

The noise vector is linearly projected upon a $4 \times 4 \times 256$ feature tensor. We assume no activation function is applied and no bias is added.

This feature tensor is then processed by a transposed convolution hidden layer. The convolutions have kernel size 3×3 and stride 2, the output is a $8 \times 8 \times 128$ feature tensor. This layer uses batch normalisation and ReLU activation. We assume the batch normalisation is applied on a per-channel basis.

This $8 \times 8 \times 128$ hidden layer is then processed by another transposed convolution layer with kernel size 3×3, stride 2, and tanh activation. No batch normalisation is applied. The output is a $16 \times 16 \times 1$ noise tensor.

The $16 \times 16 \times 1$ noise tensor is subsequently added to the $16 \times 16 \times 1$ input image resulting in a $16 \times 16 \times 1$ noisy image, which is the output of the network.

Notable is that, while network generates a complex noise pattern, the noise pattern it generates does not depend on the image. For example, the network has no way of knowing whether the person on the photo has a moustache or not, it can only decide to always add noise to the area where moustaches might be.

A.3. The adversary

The adversary, drawn in Figure A.3 is a convolutional neural network with the following layers applied in the following order:

1. Input layer $(16 \times 16 \times 1)$;
2. Convolutional layer, kernel size 3×3, stride 1, 32 filters. Per-channel batch normalisation and ReLU activation. Output format $16 \times 16 \times 32$;
3. Maxpool layer, kernel 2×2, stride 2, output format $8 \times 8 \times 32$;
4. Convolutional layer, kernel size 3×3, stride 1, 64 filters. Per-channel batch normalisation and ReLU activation. Output format $8 \times 8 \times 64$.;
5. Maxpool layer, kernel 2×2, stride 2, output format $4 \times 4 \times 64$;

6. Fully connected layer, 1024 neurons. Batch normalisation and ReLU activation, outputs a vector of size 1024;
7. Fully connected layer, 1024 neurons. Batch normalisation and ReLU activation, outputs a vector of size 1024;
8. Fully connected output layer, 2 neurons.

The output layer contains two neurons representing the adversary's belief that the subject is male/female. We assume that the output layer uses softmax activation. The softmax function applied on a vector $x \in \mathbb{R}^n$ can be computed as:

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}}.$$

A.4. Loss

We assume that we apply cross-entropy loss to the adversary. The cross entropy loss of a probability vector $z \in \mathbb{R}^n$ and one-hot label vector $y \in \{0,1\}^n$ can be computed as:

$$\text{crossentropy}(z, y) = -\sum_{i=1}^{n} y_i \ln z_i$$

A one-hot label vector is a vector which is zero in all coordinates except the coordinate representing the correct label for the sample. For example, if the neuron z_1 is the guess that the image shows a man and z_2 is the guess that the image shows a woman, the appropriate y vector for female images would be $y = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$.

The loss given to the privatiser equals minus the loss for the adversary plus a penalty based on how dissimilar the input and the output are, i.e. how much noise is added. We are given a maximum acceptable mean noise threshold T and assign extra noise every time the mean noise in a single batch exceeds T. We assume that the amount of noise is measured using the L_1 norm.

$$\text{distance penalty}(x, z) = \rho \cdot \max(||x - z||_1 - T, 0)$$

In this case, x and z are $n \times 16 \times 16 \times 1$ batches of images where n is the batch size. The operator $|| \cdot ||_1$ computes the L_1 norm of a tensor, i.e. the average of the absolute values of all its elements. The variable ρ is a given constant that tells us how heavily the distance penalty should weigh relative to the adversary's ability to interfere. It may be desirable to increase the value of ρ in later iterations of training to strongly enforce that the privatiser stays within its allowable bounds.

A.5. Training schedule

The original mentions little about the training method, so we designed our own. We train the network for 5000 epochs. During one epoch, the entire dataset is shuffled into random order, split into batches and fed into the network one time. A few images in the dataset may be skipped per epoch if we can't fit them into our constant batch size of 64. We use the Adam optimizer for both the privatiser and the adversary. We train the adversary for 10 iterations per iteration of the privatiser.

The formula for ρ we use is $1 + 0.005t$, where $t \in \mathbb{N}$ is the number of the current batch. I.e. ρ starts at 1 and increases by 0.005 for every batch we train the network with.

A.6. Criticisms on the network

We implemented the above networks to stay as close to the original article as possible. We do however not think that the networks are well designed. Based on our practical intuition, we see the following problems:

- The FFNP network is a fully connected neural network with three hidden layers. However, the current consensus seems to be that more than using more than two hidden layers for a fully connected network is not very helpful;

- The FFNP network needs to produce the entire image as output, not just the noise. This means that the network needs to spend neurons to remember what the image looks like even if it judges parts of the image to be not gender indicative. This wastes many neurons on remembering simple pixels which could otherwise have been used for pattern spotting. We think we could achieve way better results if the network only had to output the noise added to the image like the TCNNP does.

- Leaky ReLU is not a sensible activation function for the output layer because the output should lie in the range $[0, 1]$ rather than $(-\infty, \infty)$. Either an activation function with that range (such as sigmoid) should be used, or the output should be clipped.

- We think that not having access to the input image is a severe handicap for the TCNNP. For example, moustaches are gender indicative so the TCNNP can learn to add noise to the moustache area, but should it make the area lighter or darker? Ideally the area should become lighter when there is a moustache and darker where there isn't; doing the opposite won't hide anything. Since the TCNNP has no way of knowing whether there is a moustache, it can't know in which direction the noise should go.

- The adversary network contains very few convolutional layers. Usually in convolutional networks, early layers are supposed to detect basic features like edges, which are then processed into more complex features in later layers. With only two convolutional layers, the network doesn't get much of a chance to detect complex features.

- The fully connected layers in the adversary network are huge. The convolutional layers only contain $1 \cdot 3 \cdot 3 \cdot 32 + 2 \cdot 32 = 416$ and $32 \cdot 3 \cdot 3 \cdot 64 + 2 \cdot 64 = 18,496$ trainable weights respectively. On the other hand, the second fully connected hidden layers uses $1024 \cdot 1024 + 2 \cdot 1024 = 1,050,624$ weights, and the first fully connected layer uses about half of that. The training set contains only $1,740 \cdot 256 = 445,440$ pixels in total, making the adversary over three times the size of the training set, which is ridiculous.

- As far as the article mentions, no preprocessing of the dataset is applied. Basic preprocessing such as normalising each pixel to have mean zero and unit variance across the dataset is said to significantly improve neural network performance. Moreover, simple data augmentation techniques such as mirroring each image horizontally could double the amount of data available.

B. Proof of continuity

Let us assume there is a private random variable Y which is Bernouilli distributed, and there is a random variable Z which we release to the public, such that Z conditioned on $Y = 1$ is distributed like a random variable A, and Z conditioned on $Y = 0$ is distributed as a random variable B. We can then write $Z = YA + (1 - Y)B$.

We consider two kinds of measures for the amount of information Z leaks about Y. The first measure is the f-information between Y and Z, which is defined as the f-divergence between the joint distribution of (Y, Z) and the product of the marginal distributions of (Y, Z). The mutual information between Y and Z is a special case of the f-information between Y and Z where $f(x) = x \ln x$ [15].

The second measure we look at is the 1-Wasserstein distance between A and B, also referred to as the earth mover distance. The 1-Wasserstein distance computes distances between distributions, but we'll make a slight abuse of notation and also use it for distances between random variables:

$$d_W(A, B) = \inf_{(X,Y) \in \Pi(A,B)} \mathrm{E}\left[||X - Y||\right],$$

where $\Pi(A, B)$ is the set of all joint distributions whose marginal distributions are equal to those of A and B. Intuitively, this can be thought of as a transport problem where the probability mass of the distribution of A needs to be optimally transported to the probability mass of the distribution of B, and the cost of transporting mass is the amount of mass to be transported times the distance it must be transported over. The 1-Wasserstein distance between two random variables is the cost of the optimal transport plan.

In this appendix, we prove that under some conditions the f-divergences are absolutely continuous with respect to the 1-Wasserstein distance, in the sense that the f-divergence between series of random variable approaches zero whenever the 1-Wasserstein distance between them does.

Theorem 1. *Let $(A_n)_{n \in \mathbb{N}}$ and $(B_n)_{n \in \mathbb{N}}$ be series of random variables on $\mathbb{R}^k$ with continuous probability distributions, whose probability density functions $a_n, b_n : \mathbb{R}^k \to \mathbb{R}$ are L-Lipschitz continuous. If A_n converges to B_n under the 1-Wasserstein metric d_W, i.e. $\lim_{n\to\infty} d_W(A_n, B_n) = 0$, then the probability densities a_n converge uniformly to b_n as $n \to \infty$, i.e. $\lim_{n\to\infty} ||a_n - b_n||_\infty = \lim_{n\to\infty} \sup_{x \in \mathbb{R}^k} ||a_n(x) - b_n(x)|| = 0$.*

Theorem 2. *Let $(A_n)_{n \in \mathbb{N}}$ and $(B_n)_{n \in \mathbb{N}}$ be continuously distributed random variables with continuous probability density functions a_n and b_n such that a_n converges uniformly to b_n as $n \to \infty$. Let $V \sim \mathrm{Bernoulli}(\frac{1}{2})$. Assume further that for at least one of the series $(A_n)_{n \in \mathbb{N}}$, $(B_n)_{n \in \mathbb{N}}$, the probability distributions of said series are tight; i.e. in the case* *of $(A_n)_{n\in\mathbb{N}}$ it means for all $\varepsilon > 0$ there exists a compact set $K_\varepsilon \subset \mathbb{R}^k$ such that for all $n \in \mathbb{N}$ we have $\mathrm{P}(A_n \in K_\varepsilon) > 1 - \varepsilon$.*

Let $f : (0, \infty) \to \mathbb{R}$ be a convex function such that $f(1) = 0$ and $\lim_{x\to 0} f(x) < \infty$. Let $\mathcal{I}_n^f$ be the f-information between the random variables $VA_n + (1 - V)B_n$ and V. Then $\lim_{n\to\infty} \mathcal{I}_n^f = 0$.

Note must be taken that the first theorem does not state that a_n or b_n are convergent series, it merely states that the distance between $||a_n - b_n||_\infty$ gets small as $n \to \infty$.

We shall prove the first theorem after establishing some lemmas. In the following text, A_n, a_n, B_n, b_n as well as L, k shall hold the same definition and assumptions as they did in Theorem 1.

Remember that the 1-Wasserstein distance between random variables A_n and B_n is defined as

$$d_W(A_n, B_n) = \inf_{(X_n,Y_n)\in\Pi(A_n,B_n)} \mathrm{E}\left[||X_n - Y_n||\right],$$

Since the 1-Wasserstein distance is the infimum of $\mathrm{E}[||X_n - Y_n||]$ across all possible joint distributions between (A_n, B_n), we are able to find joint distributions such that $\mathrm{E}[||X_n - Y_n||] - d_W(A_n, B_n)$ is arbitrarily small.

We shall now define (X_n, Y_n) as jointly distributed random variables such that $\mathrm{E}\left[||X_n - Y_n||\right] < d_W(A_n, B_n) + 2^{-n}$, and the marginal distributions of X_n, Y_n are identical to those of A_n, B_n.

Lemma B.1. *For all $\varepsilon_c, \varepsilon_b > 0$ there exists an $N \in \mathbb{N}$ such that for all $n > N$, we have $\mathrm{P}(||X_n - Y_n|| > \varepsilon_b) < \varepsilon_c$.*

Proof. By the assumption that $\lim_{n\to\infty} d_W(A_n, B_n) = 0$ and the definition of (X_n, Y_n), it follows that

$$\begin{aligned}\lim_{n\to\infty} \mathrm{E}\left[||X_n - Y_n||\right] &\leq \lim_{n\to\infty} \left(d_W(A_n, B_n) + 2^{-n}\right) \\ &= \lim_{n\to\infty} d_W(A_n, B_n) + \lim_{n\to\infty} 2^{-n} \\ &= 0 + 0 = 0.\end{aligned}$$

Let $\varepsilon_b > 0$ arbitrarily. The Markov inequality gives us

$$0 \leq \mathrm{P}(||X_n - Y_n|| > \varepsilon_b) \leq \frac{\mathrm{E}\left[||X_n - Y_n||\right]}{\varepsilon_b}.$$

The right-hand side converges to zero as $n \to \infty$, so the squeeze theorem gives us $\lim_{n\to\infty} \mathrm{P}(||X_n - Y_n|| > \varepsilon_b) = 0$ for all $\varepsilon_b > 0$. The lemma now follows from the definition of the limit. □

Lemma B.2. *There exists a constant $M > 0$ which is an upper bound for all L-Lipschitz continuous probability density functions $f : \mathbb{R}^k \to \mathbb{R}$.*

Proof. If for some $(x, y) \in \mathbb{R}^k \times \mathbb{R}$ we have $f(x) = y$, then under the graph of f lies a $k+1$-dimensional cone $C_{x,y} \subset \mathbb{R}^k \times \mathbb{R}$ with height y and a sphere $B_n(x; y/L) \times \{0\}$ as base. The integral of f must be at least the volume of this cone. The cone's volume approaches infinity as $y \to \infty$. Since the integral of f equals 1, the volume of the cone cannot exceed 1, thus y must be bounded. □

Lemma B.3. *There exists a constant $M > 0$ such that for all measurable $U \subset \mathbb{R}^k$ and all $n \in \mathbb{N}$ we have*

$$\begin{aligned} \mathrm{P}(A_n \in U) &\leq M \cdot \lambda^k(U), \\ \mathrm{P}(B_n \in U) &\leq M \cdot \lambda^k(U), \end{aligned}$$

where λ^k is the k-dimensional Lebesgue measure.

Proof. Let M be as in Lemma B.2. Then

$$\mathrm{P}(A_n \in U) = \int_U a_n \, \mathrm{d}\lambda^k \leq \int_U \sup a_n \, \mathrm{d}\lambda^k = \lambda^k(U) \cdot \sup a_n \leq \lambda^k(U) \cdot M.$$

The same holds for all B_n. □

Lemma B.4. *Let $s > 0$, $\varepsilon > 0$. Then there exists an $N \in \mathbb{N}$ such that for all $n > N$, the following holds: if $I \subset \mathbb{R}^k$ is a cube with edge length s, i.e. some subset of the form $[x_1, x_1 + s] \times [x_2, x_2 + s] \times \cdots \times [x_k, x_k + s]$ for arbitrary $x_1, \ldots, x_k$, then:*

$$|\mathrm{P}(A_n \in I) - \mathrm{P}(B_n \in I)| < \varepsilon. \tag{B.1}$$

Proof. Let M be as in Lemma B.3. Let $\varepsilon > 0$ be arbitrary. Define $\varepsilon_c = \frac{1}{2}\varepsilon$ and choose $\varepsilon_b > 0$ such that the following equations are satisfied:

$$\begin{aligned} \varepsilon_b &< \tfrac{1}{2}s, \\ M \cdot \left((s + 2\varepsilon_b)^k - s^k\right) &< \tfrac{1}{2}\varepsilon, \\ M \cdot \left(s^k - (s - 2\varepsilon_b)^k\right) &< \tfrac{1}{2}\varepsilon. \end{aligned}$$

Note that as ε_b converges to zero, all of the left hand sides converge to zero, so an ε_b that satisfies these inequalities must exist.

Let N be as in Lemma B.1 for these values of ε_c and ε_b. We will show that Inequality (B.1) holds for all $n > N$.

Let $n > N$ arbitrary. Let I be an arbitrary cube with side s. We make use of the notion that X_n, Y_n have the same marginal distribution as A_n, B_n, hence $P(A_n \in I) = P(X_n \in I)$ and $P(B_n \in I) = P(Y_n \in I)$. We will first demonstrate the following lower bound:

$$\mathrm{P}(X_n \in I) > \mathrm{P}(Y_n \in I) - \varepsilon.$$

Note that

$$\begin{aligned} \mathrm{P}(X_n \in I) &\geq \mathrm{P}(Y_n \in I \wedge X_n \in I) \\ &= \mathrm{P}(Y_n \in I \wedge X_n \in I) + \mathrm{P}(Y_n \in I \wedge X_n \notin I) - \mathrm{P}(Y_n \in I \wedge X_n \notin I) \\ &= \mathrm{P}(Y_n \in I) - \mathrm{P}(Y_n \in I \wedge X_n \notin I). \end{aligned} \tag{B.2}$$

We will now show that $\mathrm{P}(Y_n \in I \wedge X_n \notin I) < \varepsilon$. Define I_b as the subset of I whose distance from the border of I is less than ε_b, and define $I_c = I \setminus I_b$. Because I is a cube with edge length s, the volume of I_c is that of a cube with edge length $s - 2\varepsilon_b$, which is $(s-2\varepsilon_b)^k$, and the volume of I_b is $s^k - (s-2\varepsilon_b)^k$.

Having divided I into the disjoint sets I_b and I_c, we get

$$\mathrm{P}(Y_n \in I \wedge Y_n \notin I) = \mathrm{P}(Y_n \in I_b \wedge Y_n \notin I) + \mathrm{P}(Y_n \in I_c \wedge X_n \notin I). \tag{B.3}$$

Lemma B.3 gives us:

$$\mathrm{P}(Y_n \in I_b \wedge X_n \notin I) \leq \mathrm{P}(Y_n \in I_b) \leq M \cdot \lambda^k(I_b) \leq M \cdot \left(s^k - (s-2\varepsilon_b)^k\right) < \tfrac{1}{2}\varepsilon.$$

The event $Y_n \in I_c \wedge X_n \notin I$ implies that the distance between X_n and Y_n must be at least ε_b due to the construction of I_c. Lemma B.1 gives us:

$$\begin{aligned}\mathrm{P}(Y_n \in I_c \wedge X_n \notin I) &\leq \mathrm{P}(Y_n \in I_c \wedge \|X_n - Y_n\| > \varepsilon_b)\\ &\leq \mathrm{P}(\|X_n - Y_n\| > \varepsilon_b)\\ &< \varepsilon_c = \tfrac{1}{2}\varepsilon.\end{aligned}$$

Substituting this in Equations (B.2) and (B.3) gives us

$$\begin{aligned}\mathrm{P}(Y_n \in I \wedge X_n \notin I) &< \tfrac{1}{2}\varepsilon + \tfrac{1}{2}\varepsilon = \varepsilon\\ \mathrm{P}(X_n \in I) &> \mathrm{P}(Y_n \in I) - \varepsilon.\end{aligned}$$

This concludes the lower bound. We will now prove an upper bound:

$$\mathrm{P}(X_n \in I) < \mathrm{P}(Y_n \in I) + \varepsilon.$$

We once again start out with decomposing the probability:

$$\begin{aligned}\mathrm{P}(X_n \in I) &= \mathrm{P}(X_n \in I \wedge Y_n \in I) + \mathrm{P}(X_n \in I \wedge Y_n \notin I)\\ &\leq \mathrm{P}(Y_n \in I) + \mathrm{P}(X_n \in I \wedge Y_n \notin I).\end{aligned}$$

The event $Y_n \notin I$ happens if and only if $Y_n \in I^c$ where I^c is the complement of I. Define I_b^c as the set of all points in I^c whose distance to the border of I is less than ε_b (beware: $I_b^c \neq (I_b)^c$) and define I_c^c as $I^c \setminus I_b^c$. Then

$$\mathrm{P}(X_n \in I \wedge Y_n \notin I) = \mathrm{P}(X_n \in I \wedge Y_n \in I_b^c) + \mathrm{P}(X_n \in I \wedge Y_n \in I_c^c).$$

The set I_b^c is contained in a cube with side $s + 2\varepsilon_b$ and does not contain I, so its volume is bounded by $(s+2\varepsilon)^k - s^k$. By Lemma B.3 we get

$$\mathrm{P}(X_n \in I \wedge Y_n \in I_b^c) \leq M \cdot \lambda^k(I_b^c) < M \cdot \left((s+2\varepsilon)^k - s^k\right) < \tfrac{1}{2}\varepsilon.$$

The event $X_n \in I \wedge Y_n \in I_c^c$ implies that $\|X_n - Y_n\| > \varepsilon_c$, hence by Lemma B.1 we have

$$\mathrm{P}(X_n \in I \wedge Y_n \in I_c^c) \leq \mathrm{P}(\|X_n - Y_n\| > \varepsilon_c) < \tfrac{1}{2}\varepsilon.$$

We once again use substitution to get what we want:

$$\begin{aligned} \mathrm{P}(X_n \in I \wedge Y_n \notin I) &< \tfrac{1}{2}\varepsilon + \tfrac{1}{2}\varepsilon < \varepsilon, \\ \mathrm{P}(X_n \in I) &< \mathrm{P}(Y_n \in I) + \varepsilon. \end{aligned}$$

We now have both an upper and a lower bound for $\mathrm{P}(X_n \in I) - \mathrm{P}(Y_n \in I)$, so we can conclude:

$$\begin{aligned} ||\mathrm{P}(X_n \in I) - \mathrm{P}(Y_n \in I)|| &< \varepsilon, \\ ||\mathrm{P}(A_n \in I) - \mathrm{P}(B_n \in I)|| &< \varepsilon. \end{aligned} \qquad \square$$

We are now ready to prove Theorem 1, which states that the probability density functions a_n converge uniformly to b_n.

Proof of Theorem 1. Let $\varepsilon > 0$ arbitrarily. We must show there is an $N \in \mathbb{N}$ such that for all $n > N$ and all $x \in \mathbb{R}^k$ we have $||a_n(x) - b_n(x)|| < \varepsilon$.

Define $s = \frac{1}{4}\varepsilon L^{-1}(\sqrt{k})^{-1}$. Using Lemma B.4, there exists a $N \in \mathbb{N}$ such that for all $n > N$ and for all cubes I of side s,

$$||\mathrm{P}(A_n \in I) - \mathrm{P}(B_n \in I)|| < \tfrac{1}{2}\varepsilon s^k,$$

Equivalently,

$$||s^{-k}\mathrm{P}(A_n \in I) - s^{-k}\mathrm{P}(B_n \in I)|| < \tfrac{1}{2}\varepsilon.$$

Let $x \in \mathbb{R}^k$ and $n > N$ arbitrarily; we will show that $||a_n(x) - b_n(x)|| < \varepsilon$.

Choose a cube $I \subset \mathbb{R}^k$ of side s such that $x \in I$. Because of the mean value theorem, there exists a point $z \in I$ such that $\lambda^k(I) \cdot a_n(z) = \mathrm{P}(A_n \in I)$. Since I is a k-dimensional cube of side s, it has volume s^k and diameter $s\sqrt{k}$. This gives us $a_n(z) = s^{-k}\mathrm{P}(A_n \in I)$ and $d(x, z) \leq s\sqrt{k}$. Because a_n is L-Lipschitz continuous, we also have:

$$\begin{aligned} ||a_n(x) - a_n(z)|| &\leq L \cdot d(x, z) \\ &\leq L \cdot s\sqrt{k} \\ &= L \cdot \tfrac{1}{4}\varepsilon L^{-1}\big(\sqrt{k}\big)^{-1} \cdot \sqrt{k} \\ &= \tfrac{1}{4}\varepsilon. \end{aligned}$$

This shows that $||a_n(x) - s^{-k}\mathrm{P}(A_n \in I)|| < \frac{1}{4}\varepsilon$. Using the same argument as we just used, we can show that $||b_n(x) - s^{-k}\mathrm{P}(B_n \in I)|| < \frac{1}{4}\varepsilon$ as well.

Using the triangle inequality along with the above inequalities, we get:

$$\begin{aligned} ||a_n(x) - b_n(x)|| \leq{} & ||a_n(x) - s^{-k}\mathrm{P}(A_n \in I)|| \\ & + ||s^{-k}\mathrm{P}(A_n \in I) - s^{-k}\mathrm{P}(B_n \in I)|| + ||s^{-k}\mathrm{P}(B_n \in I) - b_n(x)|| \\ <{} & \tfrac{1}{4}\varepsilon + \tfrac{1}{2}\varepsilon + \tfrac{1}{4}\varepsilon = \varepsilon. \end{aligned} \qquad \square$$

We will now move on to proving Theorem 2. From this point on, we will no longer need to assume the requirements of Theorem 1 but instead those of Theorem 2. In particular, we no longer assume a_n and b_n to be Lipschitz-continuous, merely that a_n uniformly converges to b_n. We additionally need to assume that the probability distributions of $(A_n)_{n\in\mathbb{N}}$ or $(B_n)_{n\in\mathbb{N}}$ are tight.

First we will prove a certain property of tightly distributed random variables in the following lemma. In Lemma B.6 we will use the given that $(A_n)_{n\in\mathbb{N}}$ or $(B_n)_{n\in\mathbb{N}}$ are tight along with the given that a_n converges to b_n to show that some generalisation of this property holds for both of them.

Lemma B.5. *Let $(X_n)_{n\in\mathbb{N}}$ be a series of continuous random variables on $\mathbb{R}^k$ with continuous probability density functions f_n, such that the correspond series of probability distributions is tight. Then for all $\varepsilon > 0$ there exists a $\delta_\varepsilon > 0$ such that for all $n \in \mathbb{N}$ there exists a set $S_n^\varepsilon \subset \mathbb{R}^k$ such that the following holds:*

$$\mathrm{P}(X_n \in S_n^\varepsilon) > 1 - \varepsilon,$$
$$\inf_{x\in S_n^\varepsilon} f_n(x) \geq \delta_\varepsilon.$$

Proof. Let $\varepsilon > 0$ be arbitrary. Because the probability distributions of $(X_n)_{n\in\mathbb{N}}$ are tight, we can choose a compact set $K_{\varepsilon/2} \subset \mathbb{R}^k$ such that $P(X_n \in K_{\varepsilon/2}) > 1 - \varepsilon/2$ for all $n \in \mathbb{N}$. Define $\delta_\varepsilon = \frac{1}{2}\varepsilon/\lambda^k(K_{\varepsilon/2})$. Let $n \in \mathbb{N}$ arbitrarily. Define S_n^ε as

$$S_n^\varepsilon = \{x \in \mathbb{R}^k \mid f_n(x) \geq \delta_\varepsilon\}.$$

We will now show that this choice of S_n^ε satisfies the lemma's requirements. Due to the definition of S_n^ε we clearly have $\inf_{x\in S_n^\varepsilon} f_n(x) \geq \delta_\varepsilon$, so all we have to prove is that $\mathrm{P}(X_n \in S_n^\varepsilon) > 1 - \varepsilon$, which we shall accomplish by proving that $\mathrm{P}(X_n \notin S_n^\varepsilon) < \varepsilon$:

$$\begin{aligned}
\mathrm{P}(X_n \notin S_n^\varepsilon) &= \mathrm{P}(X_n \in \{x \in \mathbb{R}^k \mid f_n(x) < \delta_\varepsilon\}) \\
&= \mathrm{P}(X_n \in \{x \in K_{\varepsilon/2} \mid f_n(x) < \delta_\varepsilon\}) + \mathrm{P}(X_n \in \{x \in \mathbb{R}^k \setminus K_{\varepsilon/2} \mid f_n(x) < \delta_\varepsilon\}) \\
&\leq \mathrm{P}(X_n \in \{x \in K_{\varepsilon/2} \mid f_n(x) < \delta_\varepsilon\}) + \mathrm{P}(X_n \in \{x \in \mathbb{R}^k \setminus K_{\varepsilon/2}\}) \\
&< \mathrm{P}(X_n \in \{x \in K_{\varepsilon/2} \mid f_n(x) < \delta_\varepsilon\}) + \tfrac{1}{2}\varepsilon \\
&= \tfrac{1}{2}\varepsilon + \int_{\{x\in K_{\varepsilon/2} | f_n(x)<\delta_\varepsilon\}} f_n(x)\, \mathrm{d}x \\
&\leq \tfrac{1}{2}\varepsilon + \int_{\{x\in K_{\varepsilon/2} | f_n(x)<\delta_\varepsilon\}} \delta_\varepsilon\, \mathrm{d}x \\
&\leq \tfrac{1}{2}\varepsilon + \int_{K_{\varepsilon/2}} \delta_\varepsilon\, \mathrm{d}x \\
&= \tfrac{1}{2}\varepsilon + \delta_\varepsilon \lambda^k(K_{\varepsilon/2}) \\
&= \varepsilon.
\end{aligned}$$

□

Lemma B.6. *Then for all $\varepsilon > 0$ there exists a $\delta_\varepsilon > 0, N \in \mathbb{N}$ such that for all $n > N$ there exists a set $S_n^\varepsilon \subset \mathbb{R}^k$ such that:*

$$\begin{aligned} &\mathrm{P}(A_n \in S_n^\varepsilon) > 1 - \varepsilon & \qquad &\mathrm{P}(B_n \in S_n^\varepsilon) > 1 - \varepsilon, \\ &\inf_{x \in S_n^\varepsilon} a_n(x) \geq \delta_\varepsilon & &\inf_{x \in S_n^\varepsilon} b_n(x) \geq \delta_\varepsilon. \end{aligned}$$

Proof. Let $\varepsilon > 0$ be arbitrary. The requirements of Theorem 2 imply that either $(A_n)_{n \in \mathbb{N}}$ or $(B_n)_{n \in \mathbb{N}}$ have tight distributions. We assume without loss of generality that $(A_n)_{n \in \mathbb{N}}$ has tight distributions. According to Lemma B.5 there exists a $\delta_\varepsilon > 0$ such that for all $n \in \mathbb{N}$ there exists a set $S_n^\varepsilon \subset \mathbb{R}^k$ such that

$$\begin{aligned} \mathrm{P}(A_n \in S_n^\varepsilon) &> 1 - \tfrac{1}{2}\varepsilon, \\ \inf_{x \in S_n^\varepsilon} a_n(x) &\geq 2\delta_\varepsilon. \end{aligned}$$

Since B_n converges uniformly to A_n, there exists an $N \in \mathbb{N}$ such that for all $n > N$ we have for all $x \in \mathbb{R}^k$:

$$|b_n(x) - a_n(x)| < \tfrac{1}{2}\varepsilon\delta_\varepsilon.$$

This choice of $\delta_\varepsilon, N, S_n^\varepsilon$ particularly implies that these S_n^ε satisfy this lemma's requirements for A_n. We will show that they satisfy the requirements for B_n as well.

Let $n > N$ be arbitrary, derive S_n^ε and let $x \in S_n^\varepsilon$ be arbitrary. We have

$$b_n(x) = a_n(x) + (b_n(x) - a_n(x)) \geq a_n(x) - |b_n(x) - a_n(x)| \geq 2\delta - \delta_\varepsilon = \delta_\varepsilon,$$

hence $\inf_{x \in S_n^\varepsilon} \geq \delta_\varepsilon$. We will finally demonstrate that $\mathrm{P}(B_n \in S_n^\varepsilon) > 1 - \varepsilon$:

$$\begin{aligned} \mathrm{P}(B_n \in S_n^\varepsilon) &= \int_{S_n^\varepsilon} b_n \,\mathrm{d}\lambda^k \\ &= \int_{S_n^\varepsilon} a_n + b_n - a_n \,\mathrm{d}\lambda^k \\ &= \int_{S_n^\varepsilon} a_n \,\mathrm{d}\lambda^k + \int_{S_n^\varepsilon} b_n - a_n \,\mathrm{d}\lambda^k \\ &= \mathrm{P}(A_n \in S_n^\varepsilon) + \int_{S_n^\varepsilon} b_n - a_n \,\mathrm{d}\lambda^k \\ &> 1 - \tfrac{1}{2}\varepsilon + \int_{S_n^\varepsilon} b_n - a_n \,\mathrm{d}\lambda^k \\ &\geq 1 - \tfrac{1}{2}\varepsilon - \left| \int_{S_n^\varepsilon} b_n - a_n \,\mathrm{d}\lambda^k \right| \\ &\geq 1 - \tfrac{1}{2}\varepsilon - \int_{S_n^\varepsilon} |b_n - a_n| \,\mathrm{d}\lambda^k \\ &\geq 1 - \tfrac{1}{2}\varepsilon - \int_{S_n^\varepsilon} \tfrac{1}{2}\varepsilon\delta_\varepsilon \,\mathrm{d}\lambda^k \\ &= 1 - \tfrac{1}{2}\varepsilon - \tfrac{1}{2}\varepsilon\delta_\varepsilon\lambda^k(S_n^\varepsilon), \end{aligned}$$

Since $1 \geq \mathrm{P}(A_n \in S_n^\varepsilon) > \delta_\varepsilon \cdot \lambda^k(S_n^\varepsilon)$, we can derive that $\lambda^k(S_n^\varepsilon) \leq 1/\delta_\varepsilon$:

$$\begin{aligned} &\geq 1 - \tfrac{1}{2}\varepsilon - \tfrac{1}{2}\varepsilon\delta_\varepsilon \cdot (1/\delta_\varepsilon) \\ &= 1 - \varepsilon. \end{aligned}$$ □

Lemma B.7. *For all $\alpha > 1$ and all $\varepsilon > 0$, there exists a $N \in \mathbb{N}$ such that for all $n > N$ there exists an area $S_n^\varepsilon \subset \mathbb{R}^k$ such that we have $\mathrm{P}(A_n \in S_n^\varepsilon) > 1 - \varepsilon$, $\mathrm{P}(B_n \in S_n^\varepsilon) > 1 - \varepsilon$, and for all $x \in S_n^\varepsilon$ we have $\alpha^{-1} b_n(x) < a_n(x) < \alpha b_n(x)$.*

Proof. Let $\alpha > 1, \varepsilon > 0$ arbitrarily. According to Lemma B.6, there exists a $\delta_\varepsilon > 0, N_1 \in \mathbb{N}$ such that for all $n > N_1$ there exists an area $S_n^\varepsilon \subset \mathbb{R}^k$ such that the following holds:

$$\begin{gathered} \mathrm{P}(B_n \in S_n^\varepsilon) > 1 - \varepsilon, \\ \inf_{x \in S_n^\varepsilon} b_n(x) \geq \delta_\varepsilon. \end{gathered}$$

Since $\lim_{n\to\infty} ||a_n - b_n||_\infty = 0$, we can choose an $N_2 \in \mathbb{N}$ such that for all $n > N_2$ and all $x \in \mathbb{R}^k$, we have

$$|a_n(x) - b_n(x)| < \min\left(\delta \cdot (\alpha - 1),\ \delta \cdot (1 - \alpha^{-1})\right). \tag{B.4}$$

Choose $N = \max(N_1, N_2)$. Let $n > N$ be arbitrary and derive S_n^ε. This choice satisfies $\mathrm{P}(B_n \in S_n^\varepsilon) > 1 - \varepsilon$. We will now show that the ratio requirements hold as well.

For all $x \in S_n^\varepsilon$, Inequality (B.4) gives us that

$$\begin{gathered} b_n(x) - \big(b_n(x) - a_n(x)\big) = a_n(x) = b_n(x) + \big(a_n(x) - b_n(x)\big), \\ b_n(x) - |a_n(x) - b_n(x)| \leq a_n(x) \leq b_n(x) + |a_n(x) - b_n(x)|, \\ b_n(x) - \delta_\varepsilon \cdot (1 - \alpha^{-1}) < a_n(x) < b_n(x) + \delta_\varepsilon \cdot (\alpha - 1). \end{gathered}$$

Since $b_n(x) \geq \delta_\varepsilon$ on S_n^ε, we can expand this to

$$\begin{aligned} b_n(x) - b_n(x) \cdot (1 - \alpha^{-1}) &\leq b_n(x) - \delta_\varepsilon \cdot (1 - \alpha^{-1}) \\ &< a_n(x) \\ &< b_n(x) + \delta_\varepsilon \cdot (\alpha - 1) \\ &\leq b_n(x) + b_n(x) \cdot (\alpha - 1). \end{aligned}$$

By rewriting the first and last inequality, this gives us

$$b_n(x) \cdot \alpha^{-1} < a_n(x) < b_n(x) \cdot \alpha.$$ □

Proof of Theorem 2. Let f be a convex function on $(0, \infty)$ such that $f(1) = 0$ and $\lim_{x\to 0^+} f(x) < \infty$. Note that this implies that f is bounded on the interval $(0, 2]$, and can be continuously extended to be bounded on the interval $[0, 2]$ with the same bound. Let $g_n(x)$ be the probability density of $VA_n + (1 - V)B_n$, $h(z)$ be the probability mass function of V, and $p_n(x, z)$ be the joint probability density/mass of $(VA_n + (1 - V)B_n, V)$.

Using these definitions, the f-information between $VA_n + (1-V)B_n$ and V can be computed as

$$\mathcal{I}_n^f = \int_{\mathbb{R}^k} \sum_{z\in\{0,1\}} g_n(x)h(z) f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right) \,\mathrm{d}x. \tag{B.5}$$

Let $\varepsilon > 0$ be arbitrary. We will show that there exists a $N > 0$ such that for all $n > N$ we have $0 \le \mathcal{I}_n^f < \varepsilon$, which will imply that $\lim_{n\to\infty} \mathcal{I}_n^f = 0$.

Because f is convex, it is continuous at 1, so there exists a $\delta > 0$ such that

$$|x-1| < \delta \Rightarrow |f(x)| < \tfrac{1}{2}\varepsilon. \tag{B.6}$$

Choose $\varepsilon^* = \frac{1}{2}\varepsilon / \sup_{y\in(0,2]} f(y)$ and $\alpha > 1$ such that $1-\delta < \alpha^{-1} < \alpha < 1+\delta$. According to Lemma B.7, we can choose an $N \in \mathbb{N}$ such that for all $n > N$ there exists a set $S_n^{\varepsilon^*} \subset \mathbb{R}^k$ with $\mathrm{P}(B_n \in S_n^{\varepsilon^*}) > 1-\varepsilon^*$ and $\mathrm{P}(A_n \in S_n^{\varepsilon^*}) > 1-\varepsilon^*$ and for all $x \in S_n^{\varepsilon^*}$ we have $\alpha^{-1}b_n(x) < a_n(x) < \alpha b_n(x)$.

Let $n > N$ be arbitrary and derive $S_n^{\varepsilon^*}$. We will show that $0 \le \mathcal{I}_n^f < \varepsilon$.

Note that the inequality $\alpha^{-1}b_n(x) < a_n(x) < \alpha b_n(x)$ is equivalent to $\alpha^{-1}a_n(x) < b_n(x) < \alpha a_n(x)$, which further tells us that for all $x \in S_n^{\varepsilon^*}$ we have

$$\alpha^{-1}a_n(x) < \tfrac{1}{2}a_n(x) + \tfrac{1}{2}\alpha^{-1}a_n(x) < \tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x) < \tfrac{1}{2}a_n(x) + \tfrac{1}{2}\alpha a_n(x) < \alpha a_n(x),$$
$$\alpha^{-1}b_n(x) < \tfrac{1}{2}\alpha^{-1}b_n(x) + \tfrac{1}{2}b_n(x) < \tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x) < \tfrac{1}{2}\alpha b_n(x) + \tfrac{1}{2}b_n(x) < \alpha b_n(x).$$

Hence for all $x \in S_n^{\varepsilon^*}$ we have

$$1-\delta < \alpha^{-1} = \frac{a_n(x)}{\alpha a_n(x)} < \frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)} < \frac{a_n(x)}{\alpha^{-1}a_n(x)} = \alpha < 1+\delta,$$
$$1-\delta < \alpha^{-1} = \frac{b_n(x)}{\alpha b_n(x)} < \frac{b_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)} < \frac{b_n(x)}{\alpha^{-1}b_n(x)} = \alpha < 1+\delta.$$

The f-information can be computed using the integral from Equation (B.5). We split this integral in two parts: we compute an upper bound of the integral over the set $S_n^{\varepsilon^*}$ and another upper bound over the set $\mathbb{R}^k \setminus S_n^{\varepsilon^*}$. We will show that on both of these sets, the integral is upper-bounded by $\frac{1}{2}\varepsilon$, which makes $\mathcal{I}_n^f$ upper-bounded by ε over the entire $\mathbb{R}^k$.

$$\mathcal{I}^f = \int_{S_n^{\varepsilon^*}} \sum_{z\in\{0,1\}} g_n(x)h(z) f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right) \,\mathrm{d}x + \int_{\mathbb{R}^k\setminus S_n^{\varepsilon^*}} \sum_{z\in\{0,1\}} g_n(x)h(z) f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right) \mathrm{d}x. \tag{B.7}$$

We start with the integral over $S_n^{\varepsilon^*}$. We can rewrite the fraction inside f in the following way:

$$\begin{aligned}
&\int_{S_n^{\varepsilon^*}} \sum_{z\in\{0,1\}} g_n(x)h(z)f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right)\,\mathrm{d}x\\
&= \int_{S_n^{\varepsilon^*}} g_n(x)h(0)f\left(\frac{p_n(x,0)}{g_n(x)h(0)}\right) + g_n(x)h(1)f\left(\frac{p_n(x,1)}{g_n(x)h(1)}\right)\,\mathrm{d}x\\
&= \int_{S_n^{\varepsilon^*}} \tfrac{1}{4}\big(a_n(x)+b_n(x)\big)f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x)+\frac{1}{2}b_n(x)}\right) + \tfrac{1}{4}\big(a_n(x)+b_n(x)\big)f\left(\frac{b_n(x)}{\frac{1}{2}a_n(x)+\frac{1}{2}b_n(x)}\right)\,\mathrm{d}x,
\end{aligned}$$

On $S_n^{\varepsilon^*}$, both of these fractions lie in the interval $(1-\delta, 1+\delta)$, which is a neighbourhood of 1; we then use Equation (B.6) to conclude that $f(\dots)$ is less than $\frac{1}{2}\varepsilon$:

$$\begin{aligned}
&\leq \int_{S_n^{\varepsilon^*}} \tfrac{1}{4}\big(a_n(x)+b_n(x)\big)\cdot\tfrac{1}{2}\varepsilon + \tfrac{1}{4}\big(a_n(x)+b_n(x)\big)\cdot\tfrac{1}{2}\varepsilon\,\mathrm{d}x\\
&= \tfrac{1}{2}\varepsilon\big(\tfrac{1}{2}\mathrm{P}(A_n\in S_n^{\varepsilon^*}) + \tfrac{1}{2}\mathrm{P}(B_n\in S_n^{\varepsilon^*})\big)\\
&\leq \tfrac{1}{2}\varepsilon.
\end{aligned}$$

On the set $\mathbb{R}^k \setminus S_n^{\varepsilon^*}$ we cannot bound the ratios $\frac{a_n}{a_n/2+b_n/2}$ and $\frac{b_n}{a_n/2+b_n/2}$ within $(1-\delta,1+\delta)$, but we can bound it within $[0,2]$: the nominator and denominator are both nonnegative, guaranteeing that it is lower bounded by zero, and the denominator is always at least half of the nominator, guaranteeing an upper bound of 2.

$$\begin{aligned}
\int_{\mathbb{R}^k\setminus S_n^{\varepsilon^*}} \sum_{z\in\{0,1\}} g_n(x)h(z)f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right)\,\mathrm{d}x &\leq \int_{\mathbb{R}^k\setminus S_n^{\varepsilon^*}} \sum_{z\in\{0,1\}} g_n(x)h(z)\sup_{y\in(0,2]} f(y)\,\mathrm{d}x\\
&= \int_{\mathbb{R}^k\setminus S_n^{\varepsilon^*}} \big(\tfrac{1}{2}a_n(x)+\tfrac{1}{2}b_n(x)\big)\sup_{y\in(0,2]} f(y)\,\mathrm{d}x\\
&= \big(\tfrac{1}{2}\mathrm{P}(A_n\in\mathbb{R}^k\setminus S_n^{\varepsilon^*}) + \tfrac{1}{2}\mathrm{P}(B_n\in\mathbb{R}^k\setminus S_n^{\varepsilon^*})\big)\sup_{y\in(0,2]} f(y)\\
&< \varepsilon^*\sup_{y\in(0,2]} f(y)\\
&= \tfrac{1}{2}\varepsilon.
\end{aligned}$$

Both parts of Equation (B.7) have been upper-bounded by $\frac{1}{2}\varepsilon$, hence the f-information is upper-bounded by ε:

$$\mathcal{I}^f = \int_{\mathbb{R}^k} \sum_{z\in\{0,1\}} g_n(x)h(z)f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right)\,\mathrm{d}x < \tfrac{1}{2}\varepsilon + \tfrac{1}{2}\varepsilon = \varepsilon.$$

From the definition of limits now follows that $\lim_{n\to\infty}\mathcal{I}_n^f = 0$. □

C. Proof of rate of convergence

Although Theorem 2 guarantees that, under some conditions, the amount of leaked information will converge to zero if the 1-Wasserstein distance between A_n and B_n converges to zero, it does not tell us what the rate of convergence is. Does the amount of leaked information converge quickly or slowly to zero when $d_W(A_n, B_n)$ does?

With the following proposition, we claim that the rate of convergence is at least $O(-\sqrt{x}\ln x)$ for certain classes of probability distributions. We have mostly proven this proposition, but our proof of this proposition relies on one assumption (Assumption C.1) about optimal solutions of transport problems which we haven't managed to prove.

Proposition 3. *Assume that Assumption C.1 is true. Let $f : (0,\infty) \to \mathbb{R}$ be a convex function such that $\lim_{x\to 0^+} f(x) < \infty$. Let A_n and B_n be continuously distributed random variables on $\mathbb{R}^k$ with probability density functions a_n and b_n such that the following holds:*

- *There is a constant $L \in \mathbb{R}$ such that a_n and b_n are L-Lipschitz continuous;*
- *There exists a set $K \subset \mathbb{R}^k$ with $\lambda^k(K) < \infty$ such that $\operatorname{supp} a_n \subset K$ and $\operatorname{supp} b_n \subset K$;*
- *The 1-Wasserstein distance $d_W(A_n, B_n)$ is sufficiently small, in particular, such that*
$$d_W(A_n, B_n) \leq \frac{1}{8L^2\lambda^k(K)}.$$

Let $Z \sim \mathrm{Bernouilli}(\frac{1}{2})$. Then there exists constants c_1, c_2 whose value depend only on f, $\lambda^k(K)$ and L, such that the f-information $\mathcal{I}_n^f$ between Z and $ZA_n + (1-Z)B_n$ is upper bounded by

$$\mathcal{I}_n^f \leq c_1 \cdot \sqrt{d_W(A_n, B_n)}\big(c_2 - \ln d_W(A_n, B_n)\big).$$

Corollary 4. *In particular, if $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ are series of random variables such that for all $n \in \mathbb{N}$, A_n and B_n satisfy the requirements of Proposition 3 with uniform values for L and $\lambda^k(K)$, then*

$$\mathcal{I}_n^f = O\big(-\sqrt{d_W(A_n, B_n)}\ln d_W(A_n, B_n)\big).$$

We will prove this proposition in two steps: first we will give an upper bound on the L_1 distance between a_n and b_n in terms of the 1-Wasserstein distance $d_W(A_n, B_n)$, and then we will give an upper bound on the leaked information in terms of the L_1 distance.

The first lemma gives a lower bound for the cost of all joint distributions P (transport plans) that have a certain form; this form being that all probability mass either stays on the same place or is moved from a surplus (source) area to a deficit (sink) area. If we can show that for all all transport plans there exists another transport plan of the given form with less or equal cost, then this lemma would give a lower bound on the 1-Wasserstein distance. We have not managed to show this, so we're taking this as an assumption instead:

Assumption C.1. *Let* P *be a joint probability distribution on* $\mathbb{R}^k \times \mathbb{R}^k$ *such that its marginal probability distributions are continuously distributed with Lipschitz-continuous probability density functions* $a, b : \mathbb{R}^k \to \mathbb{R}^k$*, i.e. for all measurable* $A \subset \mathbb{R}^k$*:*

$$\mathrm{P}(A \times \mathbb{R}^k) = \int_A a(x) \, \mathrm{d}\lambda^k(x),$$
$$\mathrm{P}(\mathbb{R}^k \times A) = \int_A b(x) \, \mathrm{d}\lambda^k(x).$$

Then there exists a probability distribution P^* *with the same marginal distributions as* P*, such that* $\mathrm{E}_{x,y\sim\mathrm{P}^*}[||x-y||] \leq \mathrm{E}_{x,y\sim\mathrm{P}}[||x-y||]$ *and* P^* *can be decomposed as*

$$\mathrm{P}^* = \mu + \nu$$

with $\mu(\{(x,y) \in \mathbb{R}^k \times \mathbb{R}^k \mid x \neq y\}) = 0$*, and for all measurable sets* $A \subset \mathbb{R}^k$ *we have*

$$\nu(A \times \mathbb{R}^k) = \int_A \max(a(x) - b(x), 0) \, \mathrm{d}\lambda^k(x)$$
$$\nu(\mathbb{R}^k \times A) = \int_A \max(b(x) - a(x), 0) \, \mathrm{d}\lambda^k(x).$$

The truth of the above assumption is necessary for the following lemma to be useful.

Lemma C.1. *Let* P *be a probability distribution on* $\mathbb{R}^k \times \mathbb{R}^k$ *such that* P *can be decomposed as*

$$\mathrm{P} = \mu + \nu$$

such that $\mu(\{(x,y) \in \mathbb{R}^k \times \mathbb{R}^k \mid x \neq y\}) = 0$ *and there exists a* $2L$*-Lipschitz continuous function* $h : \mathbb{R}^k \to \mathbb{R}$ *such that for all measurable sets* $A \subset \mathbb{R}^k$ *we have*

$$\nu(A \times \mathbb{R}^k) = \int_A \max(h(x), 0) \, \mathrm{d}\lambda^k(x), \tag{C.1}$$
$$\nu(\mathbb{R}^k \times A) = \int_A \max(-h(x), 0) \, \mathrm{d}\lambda^k(x),$$

then $\mathrm{E}_{x,y\sim\mathrm{P}}[||x-y||] \geq \frac{1}{2}||h||_2^2/(2L)^2$*.*

Proof. We start by splitting the P measure into μ and ν:

$$\begin{aligned}\mathrm{E}_{x,y\sim\mathrm{P}}[||x-y||] &= \int ||x-y|| \, \mathrm{dP}(x,y) \\ &= \int ||x-y|| \, \mathrm{d}\mu(x,y) + \int ||x-y|| \, \mathrm{d}\nu(x,y),\end{aligned}$$

Note that $x = y$ on the entire support of μ, so $\int ||x - y||\ \mathrm{d}\mu(x, y)$ equals zero:

$$= \int ||x - y||\ \mathrm{d}\nu(x, y),$$

The measure ν is supported on the set $\operatorname{supp}\nu = \{(x, y) \in \mathbb{R}^k \times \mathbb{R}^k \mid h(x) > 0, h(y) < 0\}$. We now explicitly add an indicator function of this support to the integral for clarity:

$$= \int ||x - y|| \cdot \mathbf{1}_{\operatorname{supp}\nu}(x, y)\ \mathrm{d}\nu(x, y),$$

For any x such that $h(x) > 0$, we know that $h(y) > 0$ for all $y \in B(x; h(x)/(2L))$ due to the $2L$-Lipschitz continuity of h. Since the support of ν only contains pairs (x, y) with $h(x) > 0, h(y) < 0$, we have $\mathbf{1}_{\operatorname{supp}\nu}(x, y) = 0$ for almost all all $y \in B(x; h(x)/(2L))$, and hence $\mathbf{1}_{\operatorname{supp}\nu}(x, y) \neq 0 \implies ||x - y|| \geq h(x)/(2L)$ for almost all x, y:

$$\begin{aligned} &\geq \int \frac{h(x)}{2L} \cdot \mathbf{1}_{\operatorname{supp}\nu}(x, y)\ \mathrm{d}\nu(x, y) \\ &= \int \frac{h(x)}{2L}\ \mathrm{d}\nu(x, y), \end{aligned}$$

The value of the function being integrated over is independent of y, so we can make use of the density of ν given by Equation (C.1):

$$= \int \frac{h(x)}{2L} \cdot \max(h(x), 0)\ \mathrm{d}\lambda^k(x).$$

By swapping the roles of the y and x variables and using $h^*(x) = -h(x)$, we can similarly derive the following inequality:

$$\begin{aligned} \mathrm{E}_{x,y\sim\mathrm{P}}[||y - x||] &\geq \int \frac{h^*(y)}{2L} \cdot \max(h^*(y), 0)\ \mathrm{d}\lambda^k(y) \\ &= \int \frac{h(y)}{2L} \cdot \min(h(y), 0)\ \mathrm{d}\lambda^k(y). \end{aligned}$$

We note that $||y - x|| = ||x - y||$. We now use both inequalities together to prove the lemma:

$$\begin{aligned} \mathrm{E}_{x,y\sim\mathrm{P}}[||x - y||] &= \tfrac{1}{2}\mathrm{E}_{x,y\sim\mathrm{P}}[||x - y||] + \tfrac{1}{2}\mathrm{E}_{x,y\sim\mathrm{P}}[||y - x||] \\ &\geq \frac{1}{2}\int \frac{h(x)}{2L} \cdot \max(h(x), 0)\ \mathrm{d}\lambda^k(x) + \frac{1}{2}\int \frac{h(y)}{2L} \cdot \min(h(y), 0)\ \mathrm{d}\lambda^k(y) \\ &= \frac{1}{2}\int \frac{h(x)}{2L} \cdot \frac{h(x)}{2L}\ \mathrm{d}\lambda^k(x) \\ &= \frac{1}{2} \cdot \frac{||h||_2^2}{(2L)^2}. \qquad \square \end{aligned}$$

Corollary 6. *Let A_n, B_n be two random variables which are continuously distributed with L-Lipschitz continuous probability density functions a_n, b_n, and let $d_W(A_n, B_n)$ be the 1-Wasserstein distance between A_n and B_n. Then*

$$||a_n - b_n||_2^2 \leq 2(2L)^2 d_W(A_n, B_n).$$

Proof. The 1-Wasserstein distance is defined as

$$d_W(A_n, B_n) = \inf_{\mathrm{P} \in \Pi(A_n, B_n)} E_{x,y \sim \mathrm{P}}[||x - y||]$$

where $\Pi(A_n, B_n)$ is the set of all distributions whose marginal distributions equal those of A_n and B_n. According to Assumption C.1, for any $\mathrm{P} \in \Pi(A_n, B_n)$ there is another distribution P^* with $E_{x,y\sim\mathrm{P}^*}[||x - y||] \leq E_{x,y\sim\mathrm{P}}[||x - y||]$ such that P^* admits a certain decomposition. Let $\Pi^*(A_n, B_n) \subset \Pi(A_n, B_n)$ contain all distributions with said form. Then with Assumpion C.1 we can derive

$$\inf_{\mathrm{P} \in \Pi(A_n, B_n)} E_{x,y \sim \mathrm{P}}[||x - y||] = \inf_{\mathrm{P}^* \in \Pi^*(A_n, B_n)} E_{x,y \sim \mathrm{P}^*}[||x - y||],$$

hence

$$d_W(A_n, B_n) = \inf_{\mathrm{P}^* \in \Pi^*(A_n, B_n)} E_{x,y \sim \mathrm{P}^*}[||x - y||].$$

According to Lemma C.1, the bound

$$||a_n - b_n||_2^2 \leq 2(2L)^2 E_{x,y\sim\mathrm{P}^*}[||x - y||]$$

holds for any measure P^* which admits the certain decomposition, hence

$$\begin{aligned} ||a_n - b_n||_2^2 &\leq 2(2L^2) \inf_{\mathrm{P}^* \in \Pi(A_n, B_n)} E_{x,y \sim \mathrm{P}^*}[||x - y||] \\ &= 2(2L)^2 d_W(A_n, B_n). \end{aligned}$$ □

We now want to turn this bound on the L_2 norm into a bound on the L_1 norm. For this we need to assume that a_n and b_n have a finite-measure support:

Lemma C.2. *Let a_n and b_n be two functions supported in a set $K \subset \mathbb{R}^k$ with finite measure, i.e. $\lambda^k(K) < \infty$. Then*

$$||a_n - b_n||_1^2 \leq \lambda^k(K) \cdot ||a_n - b_n||_2^2.$$

Proof. Define X as a random variable on $\mathbb{R}^k$ which is uniformly distributed over K. Then the probability density function of X is $\mathbf{1}_K/\lambda^k(K)$. We use this random variable to rewrite the norm $||a_n - b_n||_1^2$ in terms of the expected value of a random variable:

$$\begin{aligned} ||a_n - b_n||_1^2 &= \left(\int_K |a_n(x) - b_n(x)| \, \mathrm{d}x\right)^2 \\ &= \left(\lambda^k(K) \cdot \int_K \frac{\mathbf{1}_K}{\lambda^k(K)} |a_n(x) - b_n(x)| \, \mathrm{d}x\right)^2 \\ &= \big(\lambda^k(K)\big)^2 \mathrm{E}[|a_n(X) - b_n(X)|]^2, \end{aligned}$$

With Jensen's inequality follows:

$$\begin{aligned}
&\leq \left(\lambda^k(K)\right)^2 \mathrm{E}[|a_n(X) - b_n(X)|^2] \\
&= \left(\lambda^k(K)\right)^2 \int_K \frac{\mathbf{1}_K}{\lambda^k(K)} |a_n(x) - b_n(x)|^2 \, \mathrm{d}x \\
&= \lambda^k(K) \cdot \int_K |a_n(x) - b_n(x)|^2 \, \mathrm{d}x \\
&= \lambda^k(K) \cdot ||a_n - b_n||_2^2. \qquad \square
\end{aligned}$$

We now proceed to bound various formula's in terms of the L_1 norm.

Lemma C.3. *For any two probability density functions a_n, b_n and any $\alpha > 0$:*

$$\mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)} > 1 + \alpha \right) \leq \tfrac{1}{2}\alpha^{-1} ||a_n - b_n||_1.$$

Proof.

$$\begin{aligned}
&\mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)} > 1 + \alpha \right) \\
&\qquad = \mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(a_n(x) > \tfrac{1}{2}(1+\alpha)(a_n(x) + b_n(x)) \right) \\
&\qquad = \mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(-\alpha a_n(x) > \tfrac{1}{2}(1+\alpha)(b_n(x) - a_n(x)) \right) \\
&\qquad = \mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(a_n(x) < \frac{1}{2}\frac{1+\alpha}{-\alpha}(b_n(x) - a_n(x)) \right) \\
&\qquad = \mathrm{P}_{x \sim \frac{1}{2}a_n + \frac{1}{2}b_n} \left(a_n(x) < \tfrac{1}{2}\left(1+\alpha^{-1}\right)(a_n(x) - b_n(x)) \right) \\
&\qquad = \int_{\{a_n < \frac{1}{2}(1+\alpha^{-1})(a_n - b_n)\}} \tfrac{1}{2}a_n + \tfrac{1}{2}b_n \, \mathrm{d}\lambda^k \\
&\qquad = \int_{\{a_n < \frac{1}{2}(1+\alpha^{-1})(a_n - b_n)\}} a_n - \left(\tfrac{1}{2}a_n - \tfrac{1}{2}b_n\right) \mathrm{d}\lambda^k \\
&\qquad \leq \int_{\{a_n < \frac{1}{2}(1+\alpha^{-1})(a_n - b_n)\}} \tfrac{1}{2}\left(1+\alpha^{-1}\right)(a_n - b_n) - \left(\tfrac{1}{2}a_n - \tfrac{1}{2}b_n\right) \mathrm{d}\lambda^k \\
&\qquad = \int_{\{a_n < \frac{1}{2}(1+\alpha^{-1})(a_n - b_n)\}} \tfrac{1}{2}\alpha^{-1}(a_n - b_n) \\
&\qquad = \tfrac{1}{2}\alpha^{-1} \int_{\{a_n < \frac{1}{2}(1+\alpha^{-1})(a_n - b_n)\}} |a_n - b_n| \, \mathrm{d}\lambda^k \\
&\qquad \leq \tfrac{1}{2}\alpha^{-1} \int_{\mathbb{R}^k} |a_n - b_n| \, \mathrm{d}\lambda^k \\
&\qquad = \tfrac{1}{2}\alpha^{-1} ||a_n - b_n||_1. \qquad \square
\end{aligned}$$

Lemma C.4. *Let $a_n : \mathbb{R}^k \to \mathbb{R}$, $b_n : \mathbb{R}^k \to \mathbb{R}$ be two probability density functions such that $\frac{1}{2}||a_n - b_n|| \leq 1$, and let Z_n be a random variable with probability density $\frac{1}{2}a_n + \frac{1}{2}b_n$. Define X_n as*

$$X_n = \frac{a_n(Z_n)}{\frac{1}{2}a_n(Z_n) + \frac{1}{2}b_n(Z_n)},$$

then $E[|X_n - 1|] < ||a_n - b_n||_1(1 - \ln \frac{1}{2}||a_n - b_n||_1)$.

Proof. We will first show some upper bound to the expected value of X conditioned on $X > 1$:

$$\begin{aligned}
\mathrm{E}[X_n \mid X_n > 1] &= \int_0^\infty \mathrm{P}(X_n > x \mid X_n > 1)\ \mathrm{d}x \\
&= 1 + \int_1^\infty \mathrm{P}(X_n > x \mid X_n > 1)\ \mathrm{d}x \\
&= 1 + \int_1^\infty \frac{\mathrm{P}(X_n > x)}{\mathrm{P}(X_n > 1)}\ \mathrm{d}x,
\end{aligned}$$

Since the random variable X_n lies between 0 and 2, we know $\mathrm{P}(X_n > 2) = 0$ and we only need to integrate over the domain $[1, 2]$ instead of $[1, \infty)$. If we substitute x with $1 + \alpha$, we need to integrate α over the domain $[0, 1]$:

$$= 1 + \frac{1}{\mathrm{P}(X_n > 1)} \int_0^1 \mathrm{P}(X_n > 1 + \alpha)\ \mathrm{d}\alpha.$$

We now invoke the inequality from Lemma C.3:

$$\leq 1 + \frac{1}{\mathrm{P}(X_n > 1)} \int_0^1 \min\bigl(1, \tfrac{1}{2}\alpha^{-1}||a_n - b_n||_1\bigr)\ \mathrm{d}\alpha;$$

The term $\frac{1}{2}\alpha^{-1}||a_n - b_n||_1$ is less than 1 when $\alpha > \frac{1}{2}||a_n - b_n||_1$, and we use the assumption $\frac{1}{2}||a_n - b_n||_1 \leq 1$ to rewrite the integral as follows:

$$\begin{aligned}
&= 1 + \frac{1}{\mathrm{P}(X_n > 1)} \left(\int_0^{\frac{1}{2}||a_n - b_n||_1} 1\ \mathrm{d}\alpha + \int_{\frac{1}{2}||a_n - b_n||_1}^1 \tfrac{1}{2}\alpha^{-1}||a_n - b_n||_1\ \mathrm{d}\alpha \right) \\
&= 1 + \frac{1}{\mathrm{P}(X_n > 1)} \left(\tfrac{1}{2}||a_n - b_n||_1 + \tfrac{1}{2}||a_n - b_n||_1 \int_{\frac{1}{2}||a_n - b_n||_1}^1 \alpha^{-1}\ \mathrm{d}\alpha \right) \\
&= 1 + \frac{\frac{1}{2}||a_n - b_n||_1}{\mathrm{P}(X_n > 1)} \left(1 + \int_{\frac{1}{2}||a_n - b_n||_1}^1 \alpha^{-1}\ \mathrm{d}\alpha \right) \\
&= 1 + \frac{\frac{1}{2}||a_n - b_n||_1}{\mathrm{P}(X_n > 1)} \left(1 + \Bigl[\ln|\alpha|\Bigr]_{\frac{1}{2}||a_n - b_n||_1}^1 \right) \\
&= 1 + \frac{\frac{1}{2}||a_n - b_n||_1}{\mathrm{P}(X_n > 1)} \left(1 - \ln \tfrac{1}{2}||a_n - b_n||_1 \right).
\end{aligned}$$

We will now work out the version for $\mathrm{E}[X_n \mid X_n < 1]$. In this case, it is helpful to define another random variable Y_n as

$$Y_n = \frac{b_n(A_n)}{\frac{1}{2}a_n(A_n) + \frac{1}{2}b_n(A_n)}$$

and rewrite X_n in terms of Y_n to get an expression similar to the previous case:

$$\begin{aligned}\mathrm{E}[X_n \mid X_n < 1] &= 2 - \mathrm{E}[2 - X_n \mid X_n < 1] \\ &= 2 - \mathrm{E}[2 - X_n \mid 2 - X_n > 1],\end{aligned}$$

Note that $2 - X_n = Y_n$, so this expectation can be written in terms of Y_n:

$$= 2 - \mathrm{E}[Y_n \mid Y_n > 1],$$

The case for Y_n can be rewritten in the same way as we have rewritten X_n:

$$\begin{aligned}&\geq 2 - \left(1 + \frac{\frac{1}{2}||a_n - b_n||_1}{\mathrm{P}(Y_n > 1)}\left(1 - \ln \tfrac{1}{2}||a_n - b_n||_1\right)\right) \\ &= 1 - \frac{\frac{1}{2}||a_n - b_n||_1}{\mathrm{P}(X_n < 1)}\left(1 - \ln \tfrac{1}{2}||a_n - b_n||_1\right).\end{aligned}$$

We now have bounds on both $\mathrm{E}[X_n \mid X_n < 1]$ and $\mathrm{E}[X_n \mid X_n > 1]$. We use this got get a bound on $E[|X_n - 1|]$:

$$\begin{aligned}\mathrm{E}[|X_n - 1|] &= \mathrm{P}(X_n > 1)\mathrm{E}[X_n - 1 \mid X_n > 1] + \mathrm{P}(X_n < 1)\mathrm{E}[1 - X_n \mid X_n < 1] \\ &\leq \tfrac{1}{2}||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1) + \tfrac{1}{2}||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1) \\ &= ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1). \qquad \square\end{aligned}$$

Lemma C.5. *Let f be a convex functions $f : (0, \infty) \to \mathbb{R}$ such that $f(1) = 0$ and $\lim_{x \to 0^+} f(x) < \infty$. Then there exists a constant K_f such that for all probability density functions $a_n : \mathbb{R}^k \to \mathbb{R}$, $b_n : \mathbb{R}^k \to \mathbb{R}$ with $\frac{1}{2}||a_n - b_n||_1 \leq 1$ we have:*

$$\int \left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \mathrm{d}x \leq K_f \cdot ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1).$$

Proof. Note that $f(1) = 0$ and both $f(2)$ and $\lim_{x \to 0^+} f(x)$ are finite. Because f is convex, it is upper-bounded on the interval $[0, 1]$ by the line between the points $(0, \lim_{x \to 0^+} f(x))$ and $(1, 0)$, and upper bounded on the interval $[1, 2]$ by the line between the points $(1, 0)$ and $(2, f(2))$. Let $K_f \in \mathbb{R}$ be the largest absolute slope of these two lines, then $f(x)$ will be bounded on the interval $[0, 2]$ by $K_f \cdot |x - 1|$.

Let Z_n be a random variable with probability density $\frac{1}{2}a_n + \frac{1}{2}b_n$. With this random variable, the integral can be reinterpreted as an expected value:

$$\int \left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \mathrm{d}x = \mathrm{E}\left[f\left(\frac{a_n(Z_n)}{\frac{1}{2}a_n(Z_n) + \frac{1}{2}b_n(Z_n)}\right)\right].$$

If we define the random variable X_n as

$$\frac{a_n(Z_n)}{\frac{1}{2}a_n(Z_n) + \frac{1}{2}b_n(Z_n)},$$

then the expected value can be further rewritten as

$$\mathrm{E}\left[f\left(\frac{a_n(Z_n)}{\frac{1}{2}a_n(Z_n) + \frac{1}{2}b_n(Z_n)}\right)\right] = \mathrm{E}\left[f\left(X_n\right)\right].$$

Since X_n only takes values in the interval $[0,2]$ we get:

$$\begin{aligned} f(X_n) &\leq K_f \cdot |X_n - 1|, \\ E[f(X_n)] &\leq K_f \cdot E[|X_n - 1|]. \end{aligned}$$

We can use Lemma C.4 to get an upper bound on this expected value:

$$\begin{aligned} K_f \cdot E[|X_n - 1|] &\leq K_f \cdot ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1), \\ \int \left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \,\mathrm{d}x &\leq K_f \cdot ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1). \square \end{aligned}$$

Proof of Proposition 3. According to Corollary 6 and Lemma C.2:

$$\begin{aligned} ||a_n - b_n||_1^2 &\leq \lambda^k(K) \cdot ||a_n - b_n||_2^2 \\ &\leq \lambda^k(K) \cdot 2(2L)^2 d_W(A_n, B_n), \\ ||a_n - b_n||_1 &\leq 2L\sqrt{2\lambda^k(K) d_W(A_n, B_n)}. \end{aligned}$$

The value $\frac{1}{2}||a_n - b_n||_1$ is less than 1 when $d_W(A_n, B_n) \leq 1/\left(8L^2\lambda^k(K)\right)$.

Let $g_n(x)$ be the probability density of $VA_n + (1-V)B_n$, $h(z)$ be the probability mass function of Z, and $p_n(x,z)$ be the joint probability density/mass of $(VA_n + (1-V)B_n, V)$. Using these definitions, the f-information between $VA_n + (1-V)B_n$ and V can be computed as

$$\mathcal{I}_n^f = \int \sum_{z \in \{0,1\}} g_n(x)h(z) f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right) \,\mathrm{d}x,$$

We rewrite this equation similarly to how we did in the proof of Theorem 2:

$$\begin{aligned} &= \int g_n(x)h(0) f\left(\frac{p_n(x,0)}{f_n(x)g(z)}\right) + g_n(x)h(1) f\left(\frac{p_n(x,1)}{f_n(x)g(z)}\right) \,\mathrm{d}x \\ &= \int \tfrac{1}{2}\left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) + \tfrac{1}{2}\left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{b_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \,\mathrm{d}x \\ &= \frac{1}{2}\int \left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{a_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \,\mathrm{d}x \\ &\quad + \frac{1}{2}\int \left(\tfrac{1}{2}a_n(x) + \tfrac{1}{2}b_n(x)\right) f\left(\frac{b_n(x)}{\frac{1}{2}a_n(x) + \frac{1}{2}b_n(x)}\right) \,\mathrm{d}x. \end{aligned}$$

Both of these intergrals are written in the form used in Lemma C.5, which means that there exists a constant K_f such that:

$$\begin{aligned}&\leq \tfrac{1}{2}K_f \cdot ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1) + \tfrac{1}{2}K_f \cdot ||b_n - a_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1)\\ &= K_f \cdot ||a_n - b_n||_1(1 - \ln \tfrac{1}{2}||a_n - b_n||_1),\end{aligned}$$

With the known bound on $||a_n - b_n||$, this can be further upper bounded by

$$\leq K_f \cdot 2L\sqrt{2\lambda^k(K)d_W(A_n, B_n)} \cdot \left(1 - \ln\left[L\sqrt{2\lambda^k(K)d_W(A_n, B_n)}\right]\right),$$

If many of these terms are aggregated into constants c_1 and c_2, this can be written as:

$$= c_1 \cdot \sqrt{d_W(A_n, B_n)}\big(c_2 - \ln d_w(A_n, B_n)\big).$$ □

D. Counterexamples

In this appendix, we will give some counterexamples that demonstrate why the requirements of the main theorems are needed, particularly Theorem 1 and Theorem 2. The requirements are considered sufficient but not necessary requirements: it is possible for functions to not satisfy the theorem's requirements without leaking private information. However, we cannot just drop one of the requirements of the theorems without adding another requirement. The following counterexamples give some functions which satisfy all but one of the requirements of the main theorems and yet leak information.

In Section 3.4.1 we already gave one counterexample showing that the 1-Wasserstein distance gives no bound on the amount of leaked information. The example random variables were continuously distributed, but their probability density functions were not continuous. In the first counterexample, we'll give an analogue with continuous probability density functions which are not uniformly Lipschitz continuous, to demonstrate that the uniform Lipschitz continuity requirement cannot be dropped.

D.1. A continuous alternative

A continuous analogue to the functions used in Section 3.4.1 would be the following:

$$a_n(x) = \mathbf{1}_{[0,1]}(x) \cdot \big(\sin(2^n \pi x - \tfrac{1}{2}\pi) + 1\big).$$

These functions have been plotted in Figure D.1. The density functions a_n are continuous, and the random variables A_n they represent do converge to the uniform distribution on $[0, 1]$, yet leaks constant amounts of information for the same reason as in Section 3.4.1: points in the set $\{x \in [0,1] \mid a_n(x) < \frac{1}{10}\}$ are at least ten times less likely to be sampled from A_n than from the uniform distribution, and the measure of that set is constant for all n. Likewise there is a constant amount of area that is about twice as likely to be sampled from A_n than B_n. If sampled points happen to lie in those areas, then with Bayes' theorem we can make a decent guess at whether it came from A_n or B_n.

This shows that merely requiring the probability density functions to be continuous is not enough and we need a stronger requirement, such as requiring them to be uniformly Lipschitz-continuous.

D.2. Satisfying Theorem 1 but not Theorem 2

Even if A_n, B_n satisfy all the requirements for Theorem 1, it is still possible for them to leak private information, which is why Theorem 2 has an additional requirement: that either $(A_n)_{n\in\mathbb{N}}$ or $(B_n)_{n\in\mathbb{N}}$ have tight distributions.

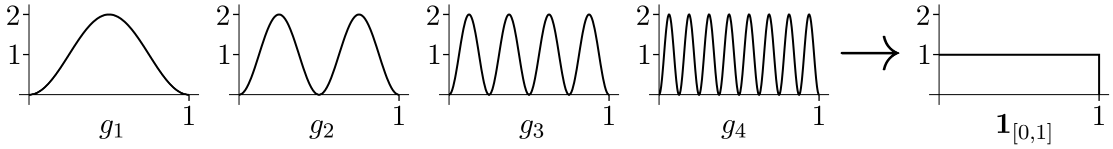


Figure D.1.: The graphs of the probability density functions $a_n(x) = \sin(2^n x - \frac{1}{4}\pi) + 1$ whose associated random variables converge to the uniform distribution on $[0, 1]$ under the 1-Wasserstein metric.

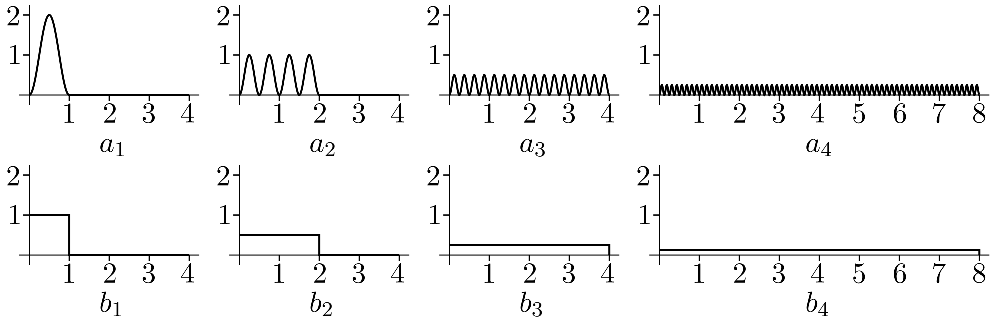


Figure D.2.: Two series probability density functions which uniformly converge to each other in the supremum-norm, but do nevertheless leak a constant amount of information.

Take a look at the following probability density functions, which have been plotted in Figure D.2:

$$\begin{aligned} a_n(x) &= \mathbf{1}_{[0,2^{n-1}]}(x) \cdot 2^{-n+1} \cdot \left(\sin(2^n \pi x - \tfrac{1}{2}\pi) + 1\right), \\ b_n(x) &= \mathbf{1}_{[0,2^{n-1}]}(x) \cdot 2^{-n+1}. \end{aligned}$$

The derivative of a_n is

$$a_n'(x) = \mathbf{1}_{[0,2^{n-1}]}(x) \cdot 2\pi \cos(2^n \pi x - \tfrac{1}{2}\pi),$$

which is bounded between $[-2\pi, 2\pi]$ for all values of n. Hence the series a_n is uniformly Lipschitz continuous and satisfies the requirements of Theorem 1. The series b_n is strictly speaking not Lipschitz continuous at the border of its support, but that could be fixed by adding a small slope at the border, which we haven't done for simplicity. Indeed, a_n and b_n do converge uniformly to each other as $\lim_{n\to\infty} ||a_n - b_n||_\infty = 0$.

Nevertheless, a combination $VA_n + (1 - V)B_n$ leaks a constant information about V like in the previous examples: there is a constant amount of probability that B_n lies in an area where A_n is much less likely to be, i.e. $\mathrm{P}(B_n \in \{x \in \mathbf{1}_{[0,2^{n-1}]} \mid a_n(x)/b_n(x) < \frac{1}{10}\})$ is independent of n.

This counterexample shows that we need some constraint on the support of the distributions A_n and B_n.

D.3. Theorem 2 does not apply to all f-informations

Theorem 2 makes a claim that applies to all f-informations whenever $\lim_{x\to 0} f(x) < \infty$. In this counterexample we will show that the theorem does not necessarily hold when $\lim_{x\to 0} f(x) = \infty$, as very similar variables my leak huge amounts of f-information.

We use the f-information with $f(x) = \frac{1}{x} - 1$ for $x > 0$, and the following series of random variables:

$$
\begin{aligned}
&A_n \sim \mathcal{N}(0,1), \\
&B_n \sim \begin{cases} \mathcal{N}(0,1) & \text{with probability } 1-4^{-n}, \\ \text{Uniform}[-2^n, 2^n] & \text{with probability } 4^{-n}, \end{cases}
\end{aligned}
$$

where $\mathcal{N}(0,1)$ refers to the standard Gaussian distribution. These variables converge uniformly to each other and are tightly distributed.

For a given sufficiently large value of n, the density function of A_n at the domain $[2^{n-1}, 2^n]$ will be of the approximate magnitude $O(e^{-n^2/2})$, whereas the the density of B_n at said domain will be of the approximate magnitude $O(2^{-4n})$. Note that the density of A_n goes to zero far quicker than the density of B_n.

Now let $V \sim \text{Bernouilli}(\frac{1}{2})$ and consider the f-information $\mathcal{I}_n^f$ between V and $VA_n + (1-V)B_n$. Let g_n be the probability density function of $VA_n + (1-V)B_n$, let h be the probability mass function of V, and let p_n be the joint density/mass function of $\big(VA_n + (1-V)B_n, V\big)$:

$$
\begin{aligned}
\mathcal{I}_n^f &= \int_{\mathbb{R}} \sum_{z\in\{0,1\}} g_n(x)h(z) f\left(\frac{p_n(x,z)}{g_n(x)h(z)}\right) \,\mathrm{d}x \\
&= \int_{\mathbb{R}} \tfrac{1}{4}\big(a_n(x)+b_n(x)\big) \left(f\left(\frac{\frac{1}{2}a_n(x)}{\frac{1}{4}a_n(x)+\frac{1}{4}b_n(x)}\right) + f\left(\frac{\frac{1}{2}b_n(x)}{\frac{1}{4}a_n(x)+\frac{1}{4}b_n(x)}\right)\right) \mathrm{d}x \\
&= \int_{\mathbb{R}} \tfrac{1}{4}\big(a_n(x)+b_n(x)\big) \left(\frac{a_n(x)+b_n(x)}{2a_n(x)} + \frac{a_n(x)+b_n(x)}{2b_n(x)} - 2\right) \mathrm{d}x \\
&= -1 + \frac{1}{4}\int_{\mathbb{R}} \big(a_n(x)+b_n(x)\big) \left(\frac{a_n(x)+b_n(x)}{2a_n(x)} + \frac{a_n(x)+b_n(x)}{2b_n(x)}\right) \mathrm{d}x \\
&\geq -1 + \frac{1}{4}\int_{[2^{n-1},2^n]} \big(a_n(x)+b_n(x)\big) \left(\frac{a_n(x)+b_n(x)}{2a_n(x)} + \frac{a_n(x)+b_n(x)}{2b_n(x)}\right) \mathrm{d}x,
\end{aligned}
$$

Since $a_n \ll b_n$ on the interval $[2^{n-1}, 2^n]$ for large values of n, we approximate that $a_n + b_n \approx b_n$:

$$
\begin{aligned}
&\approx -1 + \frac{1}{4}\int_{[2^{n-1},2^n]} \big(b_n(x)\big) \left(\frac{b_n(x)}{2a_n(x)} + \frac{b_n(x)}{2b_n(x)}\right) \mathrm{d}x \\
&= -1 + \frac{1}{4}\int_{[2^{n-1},2^n]} \frac{b_n(x)^2}{2a_n(x)} + \frac{b_n(x)}{2} \,\mathrm{d}x \\
&\geq -1 + \frac{1}{8}\int_{[2^{n-1},2^n]} \frac{b_n(x)^2}{a_n(x)} \,\mathrm{d}x.
\end{aligned}
$$

As $b_n(x)$ is of the magnitude $O(2^{-4n})$, the term $b_n(x)^2$ is of the magnitude $O(2^{-8n})$, yet a_n is of the magnitude $O(e^{-n^2/2})$. Clearly a_n converges to zero far more quickly than b_n for large values of n, so the value of the fraction $\frac{b_n(x)^2}{a_n(x)}$ explodes for large values of n. The measure of the domain of the integral also increases with increasing n, so it seems that this lower bound for $\mathcal{I}_n^f$ diverges to infinity as $n \to \infty$.

D.4. Leaking no information without satisfying Theorem 2

This counterexample exists to show that the requirements of Theorem 2 are not necessary requirements to leak no information. The counterexample is fairly simple: let P be any probability distribution on $\mathbb{R}$, then define the series $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ as:

$$\begin{aligned} A_n &\sim \mathrm{P} + n, \\ B_n &\sim \mathrm{P} + n. \end{aligned}$$

Since A_n and B_n are identically distributed, the random variable $VA_n + (1 - V)B_n$ is independent of V. The distribution P can be chosen to not be continuously distributed, and the $+n$ offsets ensure that $(A_n)_{n\in\mathbb{N}}$ nor $(B_n)_{n\in\mathbb{N}}$ is tightly distributed.

Bibliography

[1] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative adversarial privacy. *arXiv preprint arXiv:1807.05306*, 2018. First version.

[2] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679`.

[3] United States Congress. Health insurance portability and accountability act, 1996. `https://www.govinfo.gov/content/pkg/STATUTE-110/pdf/STATUTE-110-Pg1936.pdf`.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[7] Peiyuan (Alexander) Liao. The wonderful wasserstein gan. `https://medium.com/\spacefactor\@m{}liaop20/the-wonderful-wasserstein-gan-def614a8aacc`, 2018. Accessed: July 4th, 2019.

[8] Jonathan Hui. GAN – Wasserstein GAN & WGAN-GP. `https://medium.com/\spacefactor\@m{}jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490`, 2018. Accessed: July 4th, 2019.

[9] Kirti Bakshi. Wasserstein gan: An alternative to the traditional gan training. `https://www.techleer.com/articles/471-wasserstein-gan-an-alternative-to-the-traditional-gan-training/`, 2018. Accessed: July 4th, 2019.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[11] Jihun Hamm. Minimax filter: learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734, 2017.

[12] Ardhendu Tripathy, Ye Wang, and Prakash Ishwar. Privacy-preserving adversarial networks. *arXiv preprint arXiv:1712.07008*, 2017.

[13] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[14] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[15] Friedrich Liese and Igor Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[16] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[17] Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

[18] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[19] `http://mplab.ucsd.edu`. The MPLab GENKI Database.

[20] Jihun Hamm. Minimaxfilter. `https://github.com/jihunhamm/MinimaxFilter`, retrieved June 18th, 2018.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.