Population Obfuscation with GAN's

Peyton Krahulik

Dr. Kyle Caudle
Dr. Randy Hoover

South Dakota Mines

April 24, 2022

Outline

- Introduction
- Population Obfuscation
- Population Obfuscation BMI Example
- Generative Adversarial Network (GAN)
- 2 Wasserstein Distance
- Toy Two-dimensional Problem
- Experimental Design
- GAN Parameters
- Results

Introduction

- Providing real world datasets to the public without proprietary or classified information is hard.
- Data obfuscation is used to remove or obfuscate data but can be expensive.
- The five most common ways to obfuscate data are:
  1. Encryption: very secure, but useless for data analysis
  2. Tokenization: involves turning certain data entries to tokens that let authorized users see the real value
  3. Data Masking: involves substituting real values with realistic but fake values
  4. Redaction: involves permanently ”blacking” out variables
  5. Differential Privacy: Differential Privacy involves introducing noise into your dataset to maintain the privacy of the entries.

Population Obfuscation

Population obfuscation is based on the notion of a marked population $\mathcal{M}$ containing marked information. The goal is to either construct or learn the population $\mathcal{T}$. In $\mathcal{T}$ the marked observations are obfuscated, but otherwise $\mathcal{T}$ is no different from $\mathcal{M}$. For this problem we have the following assumptions:

- $\mathcal{S}_M$ is a sample from $\mathcal{M}$ and is assumed to be infinite
- $\mathcal{S}_T$ is a sample from $\mathcal{T}$ and is too small to train a generative model
- An isolation map $g : \vec{Y} \rightarrow \vec{Z} = \begin{bmatrix} \mathbf{M} \\ \mathbf{X} \end{bmatrix}$ exists, which identifies the marked information ($\mathbf{M}$) from the rest ($\mathbf{X}$)
- It is easy to mask $\mathbf{M}$ by randomizing it. It is also easy to redact $\mathbf{M}$ by changing it to a constant value
- Our target distribution for masking $\mathbf{M}$ is: $\mathcal{T} = g^{-1}(\begin{bmatrix} \mathbf{H} \\ \mathbf{X} \end{bmatrix})$, where $\mathbf{H}$ is drawn from the distribution of $\mathcal{M}$
- Our target distribution for redacting $\mathbf{M}$ is: $\mathcal{T} = g^{-1}(\begin{bmatrix} \mathbf{H}_B \\ \mathbf{X} \end{bmatrix})$, where $\mathbf{H}_B$ is a constant vector

Population Obfuscation BMI Example

Suppose we have a population $\mathcal{Y} \in \mathbb{R}^2$ which represents physical data for a patient population, y_1 is height and y_2 is weight. The task here is to obfuscate a patient's BMI ($\frac{y_2}{y_1^2}$) while preserving the patient's size ($y_1 * y_2$).

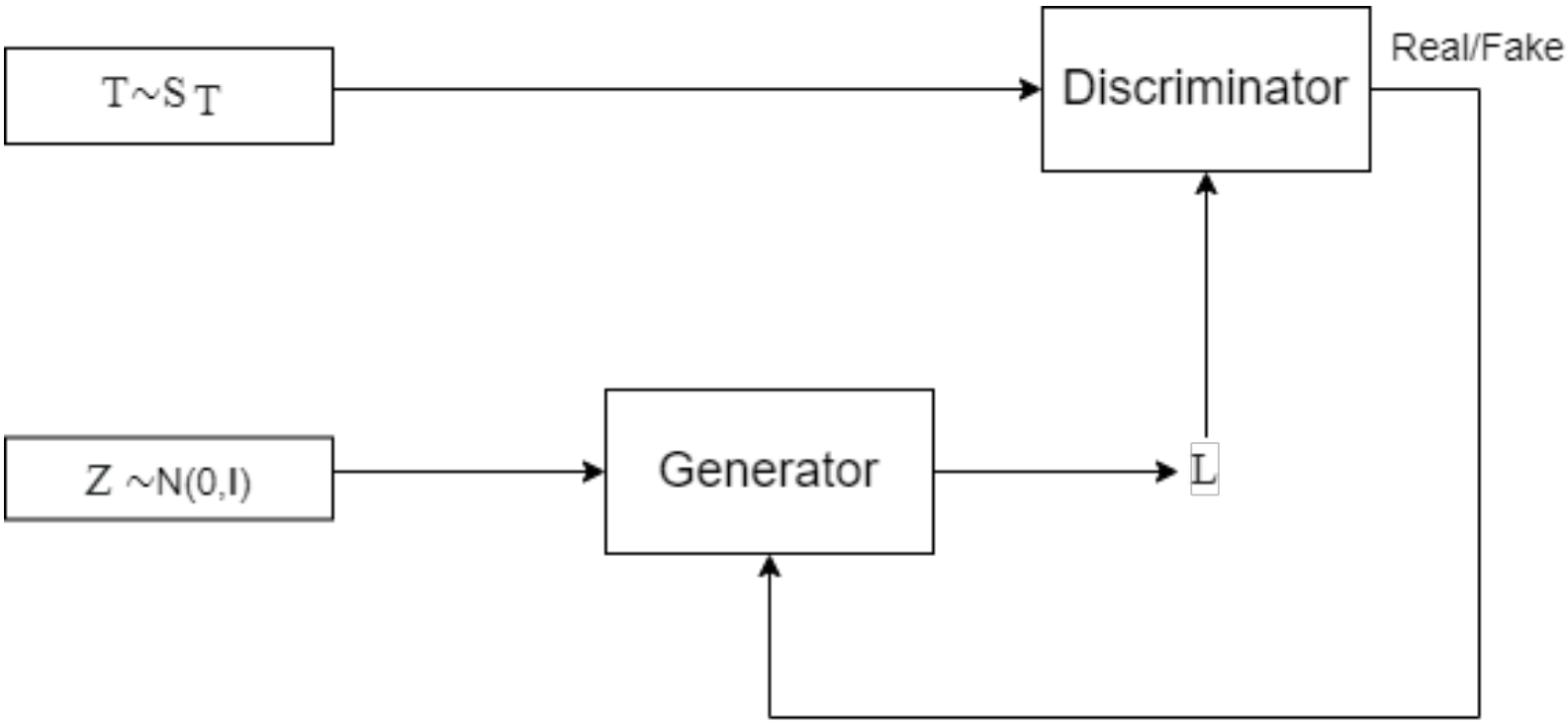
- Our isolation map is $g : \vec{Y} \to \vec{Z} = \begin{bmatrix} \mathbf{M} \\ \mathbf{X} \end{bmatrix}$), where $\mathbf{M}$ is our marked information
- In this scenario our marked information $\mathbf{M}$ = BMI. $\mathbf{M}$ has a distribution M°
- $\mathbf{M}$ and $\mathbf{X}$ are stochastically dependent
- The mark $\mathbf{M}$ is masked in $\vec{Y}$ by replacing it with $\mathbf{H} \sim M^\circ$
- $\mathbf{H}$ and $\mathbf{X}$ are stochastically independent
- The inverse transformation is $\mathcal{T} = g^{-1} \begin{bmatrix} \mathbf{H} \\ \mathbf{X} \end{bmatrix}$), where $\mathcal{T}$ is the masked target population
- Population obfuscation seeks to either generatively model $\mathcal{T}$ or to construct an estimator of the masking map from $\mathbf{M}$ to $\mathcal{T}$

Population Obfuscation BMI Example Continued

To show the whole process I'll go through a step by step example. It is important to note that we do not actually know g^{-1} in a real example.

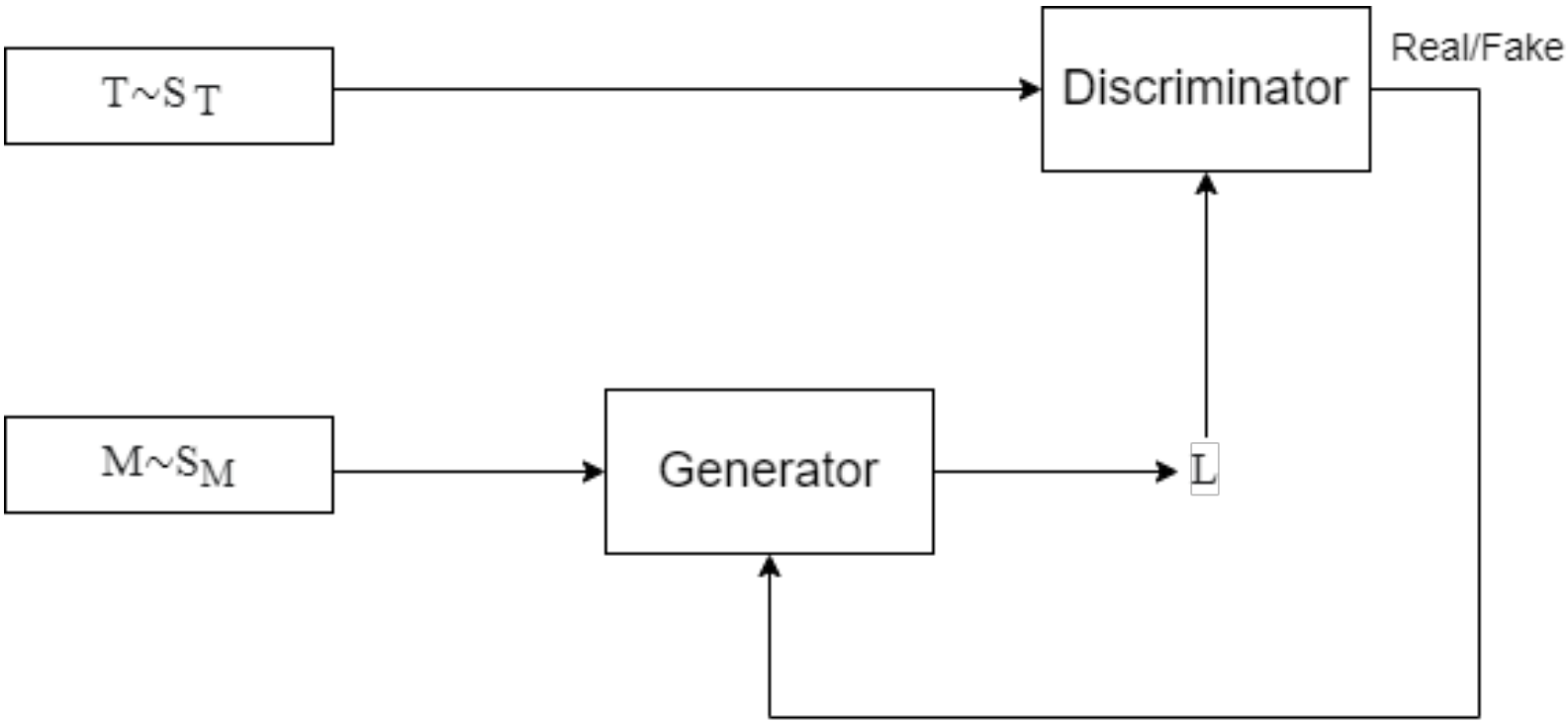
- We start with y_1 and y_2, so $\vec{Y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
- Then we transform weight and height to BMI and Size by $g(\vec{Y}) = \begin{bmatrix} y_1/y_2^2 \\ y_1 * y_2 \end{bmatrix}$
- Next we draw a sample from the distribution of $\mathcal{M}$: $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \sim M^{\circ}$
- Then the BMI is replaced by the sample: $\begin{bmatrix} h_1/h_2^2 \\ y_1 * y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{X} \end{bmatrix}$
- Finally we take the inverse transform to get a sample from $\mathcal{T}$
  $g^{-1}(\begin{bmatrix} \mathbf{H} \\ \mathbf{X} \end{bmatrix}) = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \sim \mathcal{T}$
- For our GAN we have essentially an infinite amount of samples from $\begin{bmatrix} \mathbf{M} \\ \mathbf{X} \end{bmatrix}$, and then we use the GAN to estimate $\mathcal{L}$, which is an estimate of $\mathcal{T}$

Vanilla Generative Adversarial Network



- $\mathcal{S}_{\mathcal{T}}$ is a small sample from the target population $\mathcal{T}$.
- $\mathcal{Z}$ is the standard normal distribution.
- $\mathcal{L}$ is the generator's learned distribution.

Obfuscatory Generative Adversarial Network



- $\mathcal{S}_{\mathcal{T}}$ is a small sample from the target population $\mathcal{T}$.
- $\mathcal{S}_{\mathcal{M}}$ is a large sample from the marked population $\mathcal{M}$
- $\mathcal{L}$ is the generator's learned distribution.

2 Wasserstein Distance

- The 2 Wasserstein distance between two normal distributions μ_1 and μ_2 is defined as:
- $W_2(\mu_1, \mu_2)^2 = ||m_1 - m_2||_2^2 + tr(C_1) + tr(C_2) - tr(2(C_2^{1/2} C_1 C_2^{1/2})^{1/2})$
  - μ_1 and μ_2 are distributions
  - m_1 and m_2 are vectors of means
  - C_1 and C_2 are the covariance matrices

- Imaging you have two piles of sand, the 2 Wasserstein distance measures the amount of work it takes transform one pile into a copy of the other pile.
- For the GAN, we made up an example so that we know what the target's mean and covariance matrix are.

Toy Two-dimensional Problem

To start we explored a simple two-dimensional obfuscation problem which involves a marked bivariate normal distribution $\mathcal{M}$ and seeing if we can train a GAN to transform it to into $\mathcal{T}$, which is a rotated bivariate normal distribution. Then we also train a GAN using the standard normal distribution, call it $\mathcal{Z}$, and transform it into $\mathcal{T}$ as well. This toy problem is engineered so that we know what $\mathcal{T}$ and $\mathcal{M}$ are. Some calculations yield the distributions for both $\mathcal{M}$ and $\mathcal{T}$

- $\mathcal{M} = N(\vec{0}, \Sigma_{\mathcal{M}}), \Sigma_{\mathcal{M}} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \rho & 0 \\ 0 & 1/\rho \end{bmatrix}$
- $\mathcal{T} = N(\vec{0}, \Sigma_{\mathcal{T}})$,
- $\Sigma_{\mathcal{T}} = \frac{\rho + 1/\rho}{2} I + \frac{\rho - 1/\rho}{2} \cos 2\theta \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$

Toy Two-dimensional Problem

To start we explored a simple two-dimensional obfuscation problem which involves a marked bivariate normal distribution $\mathcal{M}$ and seeing if we can train a GAN to transform it to into $\mathcal{T}$, which is a rotated bivariate normal distribution. Then we also train a GAN using the standard normal distribution, call it $\mathcal{Z}$, and transform it into $\mathcal{T}$ as well.

- $\mathcal{M} = N(\vec{0}, \Sigma_{\mathcal{M}}), \Sigma_{\mathcal{M}} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \rho & 0 \\ 0 & 1/\rho \end{bmatrix}$
- $\mathcal{Z} = N(\vec{0}, \mathbf{I}), \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Our isolation map $g : \vec{Y} \to \vec{S} = \begin{bmatrix} M \\ X \end{bmatrix}$ is the unitary transform $\tilde{\mathbf{S}} = \mathbf{U}\tilde{\mathbf{Y}}$
- $\mathbf{U}$ is the rotation matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$
- Under this transformation $Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \vec{S} = \begin{bmatrix} M \\ X \end{bmatrix}$

Toy Two-dimensional Problem Continued

- The mark M in $\vec{S}$ is $M = y_1 \cos\theta + y_2 \sin\theta$
- $\mathcal{S} = N(\vec{0}, \Sigma_{\mathcal{S}}), \Sigma_{\mathcal{S}} = \Sigma_{\mathcal{M}} + (\rho - 1/\rho)\sin\theta \begin{bmatrix} \cos\theta & -\sin\theta \\ -\sin\theta & -\cos\theta \end{bmatrix}$
- Now we introduce $\mathcal{S}^\circ$ which is the de-marked counterpart to $\mathcal{S}$. In $\mathcal{S}^\circ X$ and M are independent.
- $\mathcal{S}^\circ = N(\vec{0}, \Sigma_{\mathcal{S}^\circ}), \Sigma_{\mathcal{S}^\circ} = \Sigma_{\mathcal{M}} + (\rho - 1/\rho)\sin^2\theta \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
- $\mathbf{T} = \mathbf{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathcal{T}})$
- $\Sigma_{\mathcal{T}} = \mathbf{U}^T \Sigma_{\mathcal{S}^\circ} \mathbf{U} = \frac{\rho + 1/\rho}{2}\mathbf{I} + \frac{\rho - 1/\rho}{2}\cos 2\theta \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$
- In this example a linear map exists taking $\vec{Y} \sim \mathcal{M}$ to $\vec{T} \sim \mathcal{T}$: $g : \vec{Y} \to \vec{T} => \vec{T} = \mathbf{V}\vec{Y}$
- $\mathbf{V} = \sqrt{\Sigma_{\mathcal{T}} \Sigma_M^{-1}}$

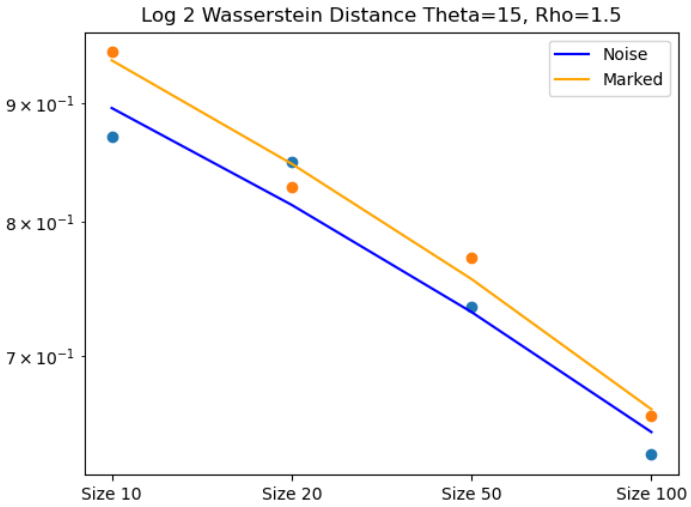
Experiment Design

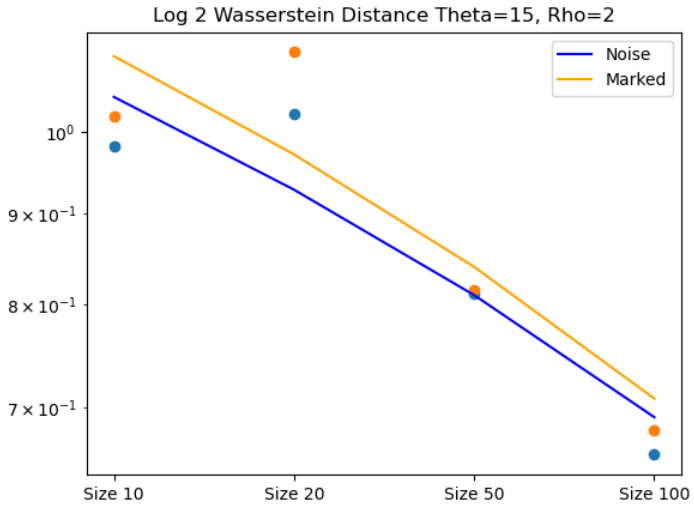
- The experiment is to run a small experiment that tests out if training a obfuscatory GAN is better then a traditional GAN with a small sample, $\mathcal{S}_{\mathcal{T}}$, from the target population $\mathcal{T}$ and a large sample, $\mathcal{S}_{\mathcal{T}}$ from the masked population $\mathcal{M}$.
- $\mathcal{S}_M = \infty, \sigma_1 * \sigma_2 = 1$
- Full factorial experiment with three experiment factor for a total of 48 design points:
  - $\rho = 1.5, 2, 2.5, 3$
  - $\theta = \frac{\pi}{12}, \frac{\pi}{6}, \frac{\pi}{4}$
  - $N_T = 10, 20, 50, 100$
- 100 runs per design point for a total of 9600 GAN's trained
- Calculate the 2 Wasserstein between $\mathcal{L}$ and $\mathcal{T}$ for both the obfuscatory GAN and the traditional GAN

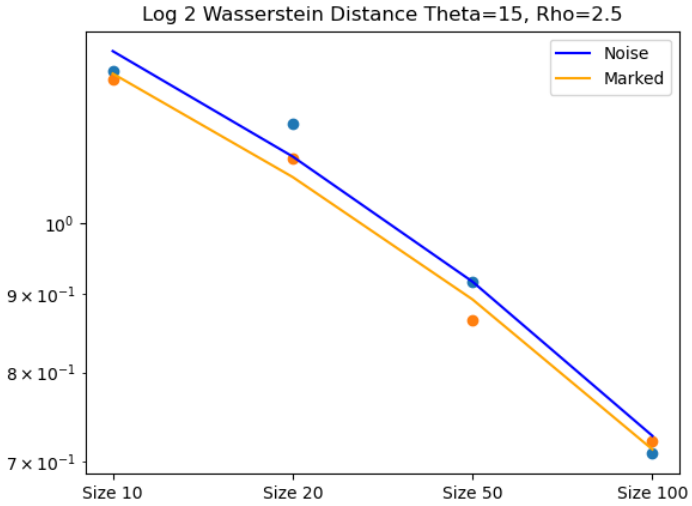
GAN Parameters

Hyperparameters	Generator	Discriminator
Layers	3	3
Nodes	128, 128, 128	32, 32, 32
Activation	Relu, Relu, Relu	Relu, Relu, Sigmoid
Optimizer	Adam	Adam
Learning Rate	0.002	0.002
Num Epochs	5000	5000
Loss Function	BCE	BCE

Results


Log 2 Wasserstein Distance Theta=15, Rho=1.5
Noise
Marked
9×10^{-1}
8×10^{-1}
7×10^{-1}
Size 10
Size 20
Size 50
Size 100


Log 2 Wasserstein Distance Theta=15, Rho=2
Noise
Marked
10^{0}
9×10^{-1}
8×10^{-1}
7×10^{-1}
Size 10
Size 20
Size 50
Size 100


Log 2 Wasserstein Distance Theta=15, Rho=2.5
Noise
Marked
10^{0}
9×10^{-1}
8×10^{-1}
7×10^{-1}
Size 10
Size 20
Size 50
Size 100

Questions?

Questions?

This project was funded by the National Institute of Standards and Technology (NIST)
Federal Award ID Number 70NANB22H020