Generative Adversarial Networks for Social Scientists

Social Science Data Lab

Marcel Neunhoeffer [1]

November 04, 2020

[1]University of Mannheim

These Cats Do Not Exist!

Figure 1: Cats from thiscatdoesnotexist.com

These Persons Do Not Exist!

Figure 2: Persons from thispersondoesnotexist.com

These Voices Never Said These Words!

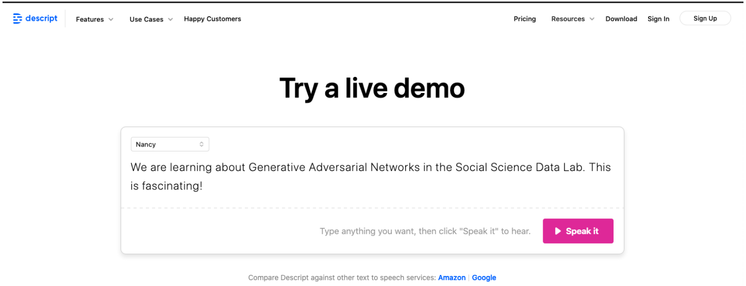


Figure 3: Voices from descript.com

Why We Should Care!

Fake Persons are Used to Spread Misinformation.

"How a fake persona laid the groundwork for a Hunter Biden conspiracy deluge" (NBC News, Oct. 29, 2020)

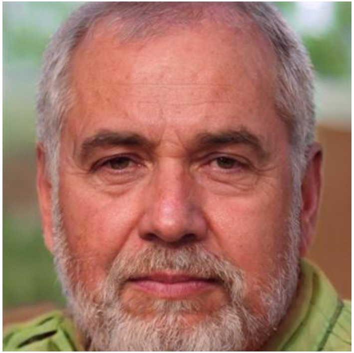


Figure 4: " 'Martin Aspen', a fake identity whose profile picture was created by artificial intelligence."

Deep Fakes Can Make the Detection of Misinformation Really Hard.

Can we still trust audio visual sources?

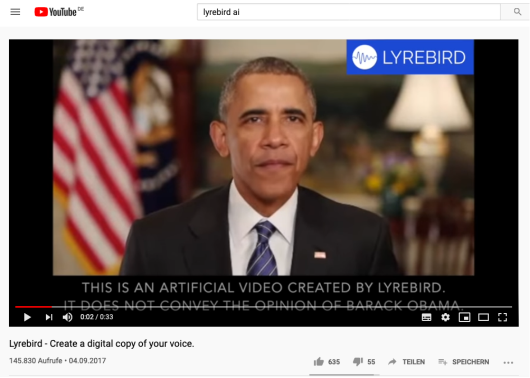


Figure 5: A Deep Fake (three years ago...).

Clever Applications of GANs Enhance our Social Science Methods Toolkit.

What I will talk about today:

- GANs for Multiple Imputation.
- GANs for Small Area Estimation
- GANs for Privacy Protective Synthetic Data
- ...

A Quick Introduction to Generative Adversarial Networks.

Generative Adversarial Nets Are Surprisingly Simple.

- Generative Adversarial Nets (GANs), introduced by Goodfellow et al. (2014), allow it to sample from arbitrary joint (continuous) distributions.
- At its core, a GAN is a minimax game with two competing actors–a discriminator (D) trying to tell real from synthetic samples and a generator (G) to produce realistic synthetic samples from random noise.
- Formally, this two-player minimax game can be written as:

$$\min_G \max_D \; \mathbb{E}_{x\sim p_X}[f(D(x))] + \mathbb{E}_{z\sim p_z}[f(1 - D(G(z)))] \tag{1}$$

where $f : [0,1] \to \mathbb{R}$ is a monotone function. For example, in standard GAN, $f(a) = \log(a)$, and in Wasserstein GAN (Arjovsky et al., 2017), $f(a) = a$.
$p_{data}(x)$ is the distribution of the real data, X is a sample from $p_{data}(x)$. The generator network $G(z)$ takes as input z from $p(z)$, where z is a random sample from a probability distribution $p(z)$.

Generative Adversarial Nets Are Surprisingly Simple.

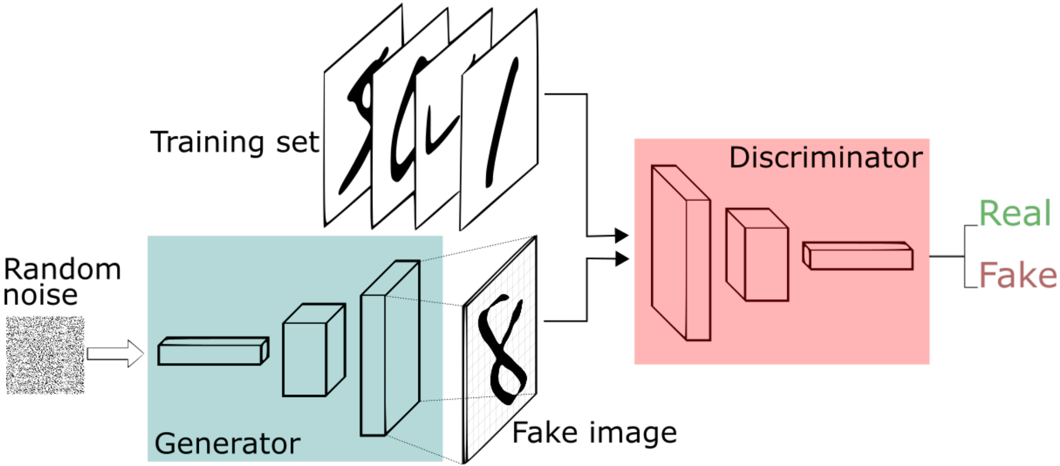


Figure 6: The architecture of a GAN. Source: Freecodecamp.org, Thalles Silva

Let’s Look at Some Code.

Social Science Applications of GANs.

Multiple Imputation with Generative Adversarial Nets.

- For imputation we only want to sample imputations for missing values from the underlying joint distribution.
- Yoon et al. (2018) and Li et al. (2019) propose a straightforward extensions to the basic GAN architecture for imputation.

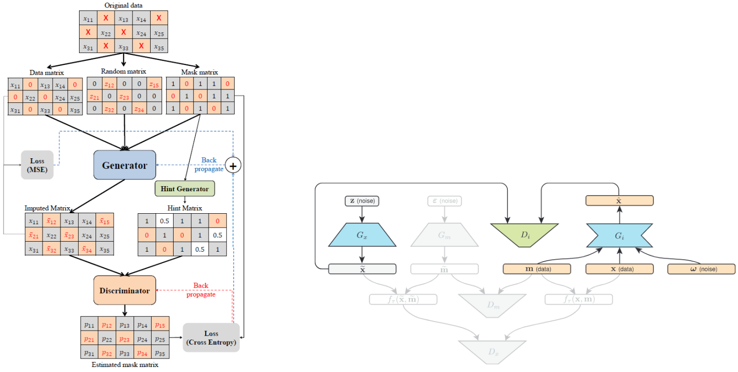


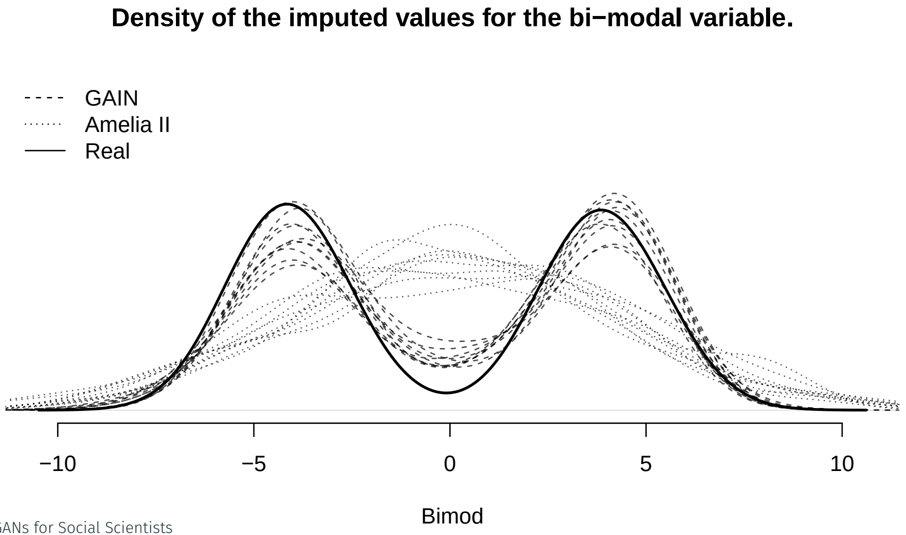
Figure 7: GAN architectures for Imputation. Left: Yoon et al., 2018, Right: Li et al., 2019.

Table 1: Overview of one generated data set for experiment 1

Variable	Type	Missing	Description
Y	Continuous	0	Dependent Variable
X	Continuous	320	Independent Variable 1 for the regression.
Bin	Binary	317	Independent Variable 2 for the regression.
Bimod	Continuous	270	Bi-modal variable.
Categ	Unordered Categorical	271	Categorical Variable.

Y is generated by: $Y = X + 2 \cdot \text{Bin} + 3 \cdot \text{Bin} \cdot X + \epsilon$.

Results for GAIN.



	obs	s	d	y		obs	s	d	y	
complete survey data	1	1	0	3		1	1	0	3	
	2	1	1	1		2	1	1	1	
	3	1	1	2		3	1	1	2	
	4	2	0	3		4	2	0	3	
	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	
	$n-m-2$	k	0	2		$n-m-2$	k	0	2	
	$n-m-1$	k	1	1		$n-m-1$	k	1	1	
	$n-m$	k	1	3		$n-m$	k	1	3	
incomplete survey data	$n-m+1$	1	NA	2		$n-m+1$	1	0.2	2	
	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	
	$n-1$	k	NA	1	$\Rightarrow$	$n-1$	k	0.9	1	
	n	k	NA	3		n	k	0.8	3	
Census data	$n+1$	1	0	NA		$n+1$	1	0	1.6	
	$n+2$	1	1	NA		$n+2$	1	1	2.1	$\bar{y}_{s=1} = 1.98$
	$n+3$	1	1	NA		$n+3$	1	1	2.0	
	$n+4$	1	1	NA		$n+4$	1	1	2.2	
	$n+5$	2	0	NA		$n+5$	2	0	1.2	
	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	
	$n+N-4$	$k-1$	1	NA		$n+N-4$	$k-1$	1	1.3	
	$n+N-3$	k	0	NA		$n+N-3$	k	0	1.9	
	$n+N-2$	k	1	NA		$n+N-2$	k	1	2.3	$\bar{y}_{s=k} = 2.05$
	$n+N-1$	k	1	NA		$n+N-1$	k	1	2.1	
	$n+N$	k	1	NA		$n+N$	k	1	2.2	

Figure 9: Multiple Imputation for SAE. Source: Honaker & Plutzer 2016.

$$
\begin{pmatrix}
obs & s & d & y \\
1 & 1 & 0 & 3 \\
2 & 1 & 1 & 1 \\
3 & 1 & 1 & 2 \\
4 & 2 & 0 & 3 \\
\vdots & \vdots & \vdots & \\
n\text{–}m\text{–}2 & k & 0 & 2 \\
n\text{–}m\text{–}1 & k & 1 & 1 \\
n\text{–}m & k & 1 & 3
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
obs & s & d & \hat{y} \\
1 & 1 & 0 & 2.7 \\
2 & 1 & 1 & 1.9 \\
3 & 2 & 0 & 1.2 \\
4 & 2 & 1 & 1.7 \\
5 & 3 & 0 & 1.5 \\
6 & 3 & 1 & 2.3 \\
\vdots & \vdots & \vdots & \vdots \\
2k\text{–}3 & k\text{–}1 & 0 & 2.1 \\
2k\text{–}2 & k\text{–}1 & 1 & 1.4 \\
2k\text{–}1 & k & 0 & 1.9 \\
2k & k & 1 & 2.6
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
obs & s & \hat{y} & w & \\
\left.\begin{matrix} 1 \\ 2 \end{matrix}\right. & \begin{matrix} 1 \\ 1 \end{matrix} & \begin{matrix} 2.7 \\ 1.9 \end{matrix} & \left.\begin{matrix} 4 \\ 7 \end{matrix}\right\} & \bar{y}_{s=1} = 2.19 \\
\begin{matrix} 3 \\ 4 \end{matrix} & \begin{matrix} 2 \\ 2 \end{matrix} & \begin{matrix} 1.2 \\ 1.7 \end{matrix} & \left.\begin{matrix} 3 \\ 9 \end{matrix}\right\} & \bar{y}_{s=2} = 1.58 \\
\begin{matrix} 5 \\ 6 \end{matrix} & \begin{matrix} 3 \\ 3 \end{matrix} & \begin{matrix} 1.5 \\ 2.3 \end{matrix} & \left.\begin{matrix} 3 \\ 6 \end{matrix}\right\} & \bar{y}_{s=3} = 2.03 \\
\vdots & \vdots & \vdots & \vdots & \\
\begin{matrix} 2k\text{–}3 \\ 2k\text{–}2 \end{matrix} & \begin{matrix} k\text{–}1 \\ k\text{–}1 \end{matrix} & \begin{matrix} 2.1 \\ 1.4 \end{matrix} & \left.\begin{matrix} 2 \\ 8 \end{matrix}\right\} & \bar{y}_{s=k-1} = 1.40 \\
\begin{matrix} 2k\text{–}1 \\ 2k \end{matrix} & \begin{matrix} k \\ k \end{matrix} & \begin{matrix} 1.9 \\ 2.6 \end{matrix} & \left.\begin{matrix} 4 \\ 7 \end{matrix}\right\} & \bar{y}_{s=k} = 2.36 \\
 & & \multicolumn{2}{c}{\sum w = N} &
\end{pmatrix}
$$

Figure 10: MRP for SAE. Source: Honaker & Plutzer 2016.

Results of GAIN for Small Area Estimation.

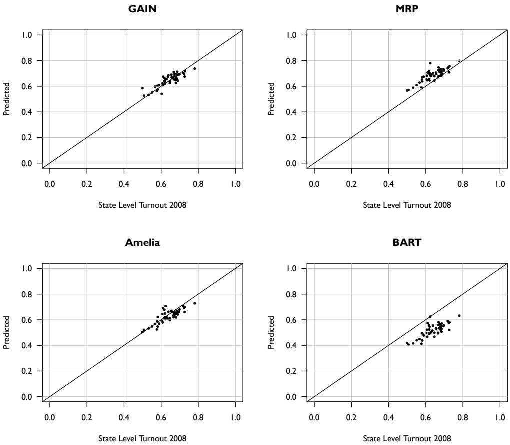


Figure 11: State Level Estimates of Turnout in the US Election 2008.

Results of GAIN for Small Area Estimation.

Table 2: RMSE and Correlation of State Level Predictions and Turnout 2008.

	RMSE	Correlation
GAIN	**0.0303**	**0.8514**
MRP	0.0561	0.8375
Amelia	0.0343	0.8286
BART	0.1280	0.7774

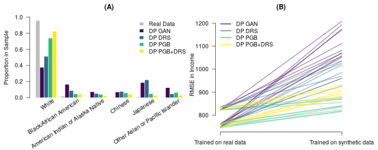
Private Post-GAN Boosting for Synthetic Data. (Neunhoeffer, Wu & Dwork)

Motivation

- A recent line of work (Beaulieu-Jones et al., 2019; Xie et al., 2018; Yoon et al., 2019) studies how one can generate synthetic data by incorporating differential privacy into *generative adversarial networks* (GANs) (Goodfellow et al., 2014).
- Due to the noise, the **convergence of GANs becomes even more elusive**. This often leads to poor utility at the end of training.
- We propose *Private post-GAN boosting (Private PGB)*, a differentially private method that **combines samples produced by the sequence of generators** during GAN training to create a **high-quality synthetic dataset**.
- We leverage the **Private Multiplicative Weights method** (Hardt & Rothblum, 2010; Hardt et al., 2012) to reweight generated samples.

Results

- Real samples from 25 multivariate normal distributions, synthetic examples without privacy from a GAN and Non-Private PGB, and synthetic examples from a GAN with differential privacy and Private PGB.



- Specific Utility of Synthetic 1940 American Census Data. Panel (A): Distribution of Race Membership in Synthetic Samples. Panel (B): Regression RMSE with Synthetic Samples.

Algorithm

Require: a private dataset $X \in \mathcal{X}^n$, a synthetic dataset B generated by the set of generators $\mathcal{G}$, a collection of discriminators $\{D_1, \ldots, D_N\}$, number of iterations T, per-round privacy budget ϵ_0, learning rate parameter η.

Initialize ϕ^1 to be the uniform distribution over B

for $t = 1, \ldots, T$ **do**

Distinguisher player: Run exponential mechanism $\mathcal{M}_E$ to select a discriminator D^t using quality score $q(X, D_j) = U(\phi^t, D_j)$ and privacy parameter ϵ_0.

Synthetic data player: Multiplicative weights update on the distribution over B: for each example $b \in B$:

$$\phi^{t+1}(b) \propto \phi^t(b) \exp(\eta D^t(b))$$

Let $\overline{D}$ be the discriminator defined by the uniform average over the set $\{D^1, \ldots, D^T\}$, and $\overline{\phi}$ be the distribution defined by the average over the set $\{\phi^1, \ldots, \phi^T\}$

Thank you for your attention!

Any Questions?

References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017. URL http://arxiv.org/abs/1701.07875.

Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P. Bhavnani, James Brian Byrd, and Casey S. Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019. doi: 10.1161/CIRCOUTCOMES.118.005122. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCOUTCOMES.118.005122.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2672–2680, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969125.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pp. 61–70, 2010. doi: 10.1109/FOCS.2010.85. URL https://doi.org/10.1109/FOCS.2010.85.

Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual*

Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., pp. 2348–2356, 2012. URL `http://papers.nips.cc/paper/4548-a-simple-and-practical-algorithm-for-differentially-private-data-release`

Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks, 2019.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018. URL `http://arxiv.org/abs/1802.06739`.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing Data Imputation using Generative Adversarial Nets. 2018. URL `http://arxiv.org/abs/1806.02920`.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1zk9iRqF7`.