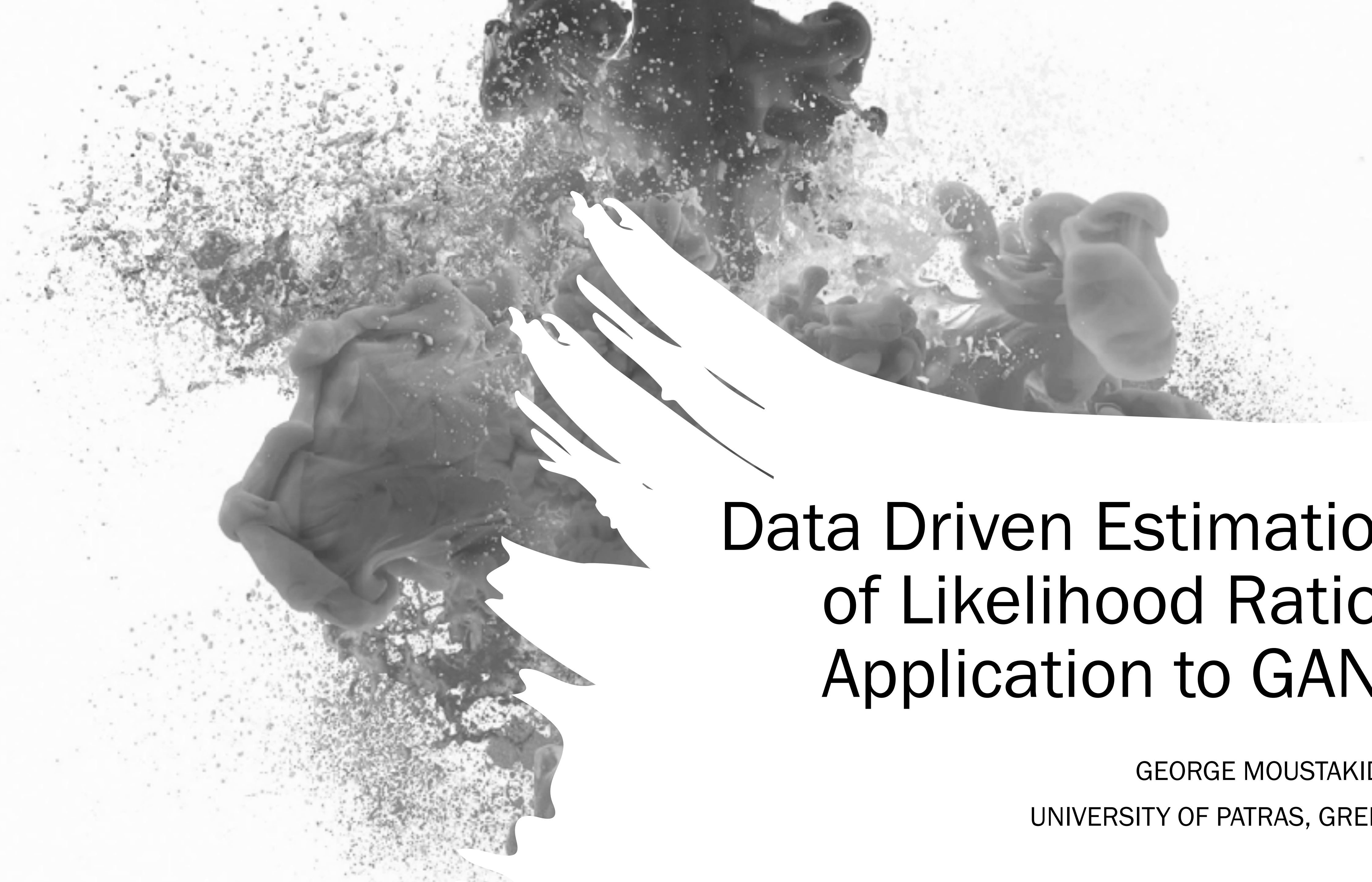


Data Driven Estimation of Likelihood Ratios Application to GANs

GEORGE MOUSTAKIDES

UNIVERSITY OF PATRAS, GREECE

Outline

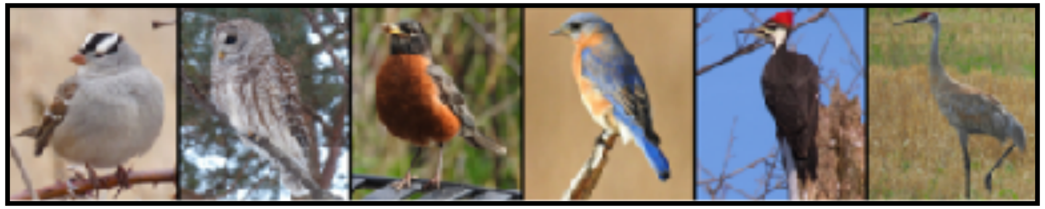
- Why likelihood ratios ?
- Data driven estimation of likelihood ratios
- Generative models
  - Design using likelihood ratio estimation
  - Generative models vs probability densities
  - Application to inverse problems

Why Likelihood Ratios ?

Random quantity

Realizations

Probability density

X:  $\cdots$  $\mathrm{f}(X)$

X_1 X_2 $\cdots$ $\cdots$ $\cdots$ X_n

X_o 

For X_o, $\mathrm{f}(X_o)$ expresses the likelihood that X_o corresponds to a bird

X:  $\cdots$  $\mathrm{f}(X)$

X_1 X_2 X_n

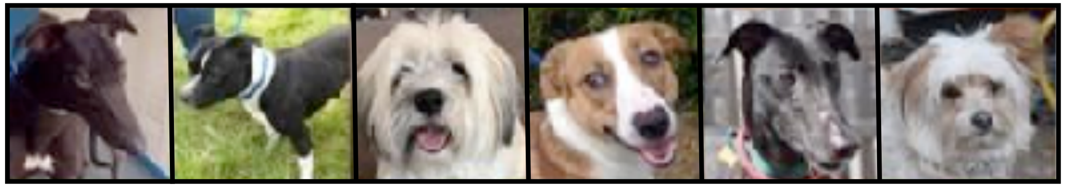
Y:  $\cdots$  $\mathrm{g}(Y)$

Y_1 Y_2 Y_m

Statistical similarity ? $\mathrm{f}(X) \overset{?}{=} \mathrm{g}(X) \Leftrightarrow \frac{\mathrm{g}(X)}{\mathrm{f}(X)} \overset{?}{=} 1$

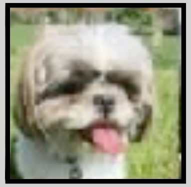
X:  $\cdots$  $f(X)$ Cat

X_1 X_2 $\cdots$ X_n

Y:  $\cdots$  $g(Y)$ Dog

Y_1 Y_2 $\cdots$ Y_m

New unlabeled data

X_0 

Decide cat or dog

Decision making
Hypothesis testing
Classification

Optimum Likelihood Ratio Test

$$\frac{g(X_0)}{f(X_0)} \underset{\text{Decide cat}}{\overset{\text{Decide dog}}{\gtreqless}} \lambda$$

Data Driven Estimation of Likelihood Ratios

THEOREM: Assume X follows density $\mathsf{f}(X)$, Y follows density $\mathsf{g}(Y)$.

Select strictly increasing function $\omega(\mathsf{r})$ and strictly positive function $\rho(z)$.
Compute $\phi(z)$, $\psi(z)$ from

$$\phi'(z) = \rho(z), \qquad \psi'(z) = -\omega^{-1}(z)\,\rho(z)$$

For $\mathsf{D}(X)$ arbitrary function define cost

$$\mathsf{J}(\mathsf{D}) = \mathbb{E}_{\mathsf{f}}\big[\phi\big(\mathsf{D}(X)\big)\big] + \mathbb{E}_{\mathsf{g}}\big[\psi\big(\mathsf{D}(Y)\big)\big]$$

Then the solution to optimization

$$\max_{\mathsf{D}} \mathsf{J}(\mathsf{D}) = \max_{\mathsf{D}} \Big\{\mathbb{E}_{\mathsf{f}}\big[\phi\big(\mathsf{D}(X)\big)\big] + \mathbb{E}_{\mathsf{g}}\big[\psi\big(\mathsf{D}(Y)\big)\big]\Big\} \;\Rightarrow\; \mathsf{D}_{\mathsf{o}}(X) = \omega\left(\frac{\mathsf{g}(X)}{\mathsf{f}(X)}\right)$$

Examples of functions

$$\omega(\mathrm{r}) = \mathrm{r} \;\; \text{(estimate likelihood ratio)}, \;\; \text{if select} \;\; \rho(z) = 1$$

$$\phi(z) = -\frac{z^2}{2}, \quad \psi(z) = z$$

Mean Square

$$\omega(\mathrm{r}) = \log \mathrm{r} \;\; \text{(estimate log-likelihood ratio)}, \;\; \text{if select} \;\; \rho(z) = e^{-0.5z}$$

$$\phi(z) = -2e^{0.5z}, \quad \psi(z) = -2e^{-0.5z}$$

Exponential

$$\omega(\mathrm{r}) = \frac{\mathrm{r}}{1+\mathrm{r}} \;\; \text{(estimate posterior probability)}, \;\; \text{if select} \;\; \rho(z) = \frac{1}{z}$$

$$\phi(z) = \log(1-z), \quad \psi(z) = \log(z)$$

Cross Entropy

Data Driven Implementation

$$\mathsf{J}(\mathsf{D}) = \mathbb{E}_{\mathsf{f}}\big[\phi\big(\mathsf{D}(X)\big)\big] + \mathbb{E}_{\mathsf{g}}\big[\psi\big(\mathsf{D}(Y)\big)\big]$$

$\{X_1, X_2, \ldots, X_n\}$ following $\mathsf{f}(X)$, $\{Y_1, Y_2, \ldots, Y_m\}$ following $\mathsf{g}(Y)$

Approximate: $\mathsf{D}(X)$ with neural network $\mathsf{D}(X,\vartheta)$, Expectations with sample means

$$\max_{\vartheta} \mathsf{J}(\vartheta) = \max_{\vartheta}\Big\{\frac{1}{n}\sum_{i=1}^{n}\phi\big(\mathsf{D}(X_i,\vartheta)\big) + \frac{1}{m}\sum_{j=1}^{m}\psi\big(\mathsf{D}(Y_j,\vartheta)\big)\Big\} \Rightarrow \vartheta_{\mathsf{o}} \Rightarrow \mathsf{D}(X,\vartheta_{\mathsf{o}})$$

We expect $\mathsf{D}(X,\vartheta_{\mathsf{o}}) \approx \mathsf{D}_{\mathsf{o}}(X) = \omega\left(\frac{\mathsf{g}(X)}{\mathsf{f}(X)}\right) \Rightarrow \omega^{-1}\big(\mathsf{D}(X,\vartheta_{\mathsf{o}})\big) \approx \frac{\mathsf{g}(X)}{\mathsf{f}(X)}$

Different $\omega(\mathsf{r})$, $\phi(z)$, $\psi(z)$ produce approximation of different quality

Summary

$\{X_1, X_2, \ldots, X_n\}$ following $\mathsf{f}(X)$, $\{Y_1, Y_2, \ldots, Y_m\}$ following $\mathsf{g}(Y)$

Select $\omega(\mathsf{r}), \rho(z)$, compute $\phi(z), \psi(z)$

$$\max_{\vartheta}\Big\{\frac{1}{n}\sum_{i=1}^{n}\phi\big(\mathsf{D}(X_i,\vartheta)\big)+\frac{1}{m}\sum_{j=1}^{m}\psi\big(\mathsf{D}(Y_j,\vartheta)\big)\Big\}$$

$$\Rightarrow\ \vartheta_{\mathsf{o}}\ \Rightarrow\ \mathsf{D}(X,\vartheta_{\mathsf{o}})\ \Rightarrow\ \omega^{-1}\big(\mathsf{D}(X,\vartheta_{\mathsf{o}})\big)\approx\frac{\mathsf{g}(X)}{\mathsf{f}(X)}$$

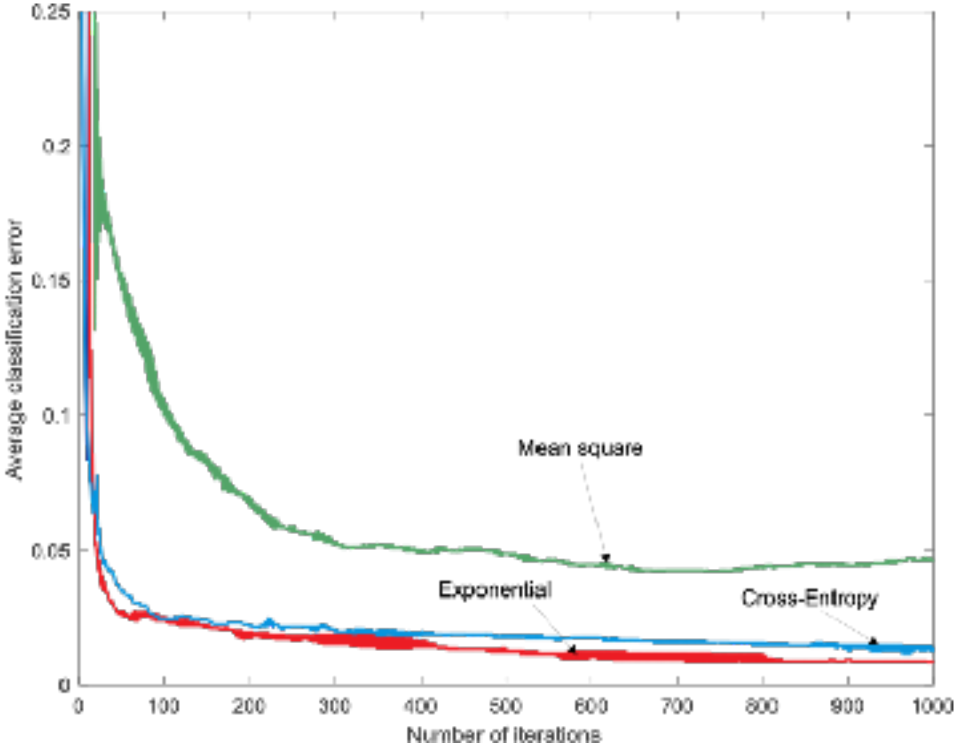
Compare $\omega^{-1}\big(\mathsf{D}(X,\vartheta_{\mathsf{o}})\big)$ to 1 to assess whether the two datasets have the same statistical behavior or not

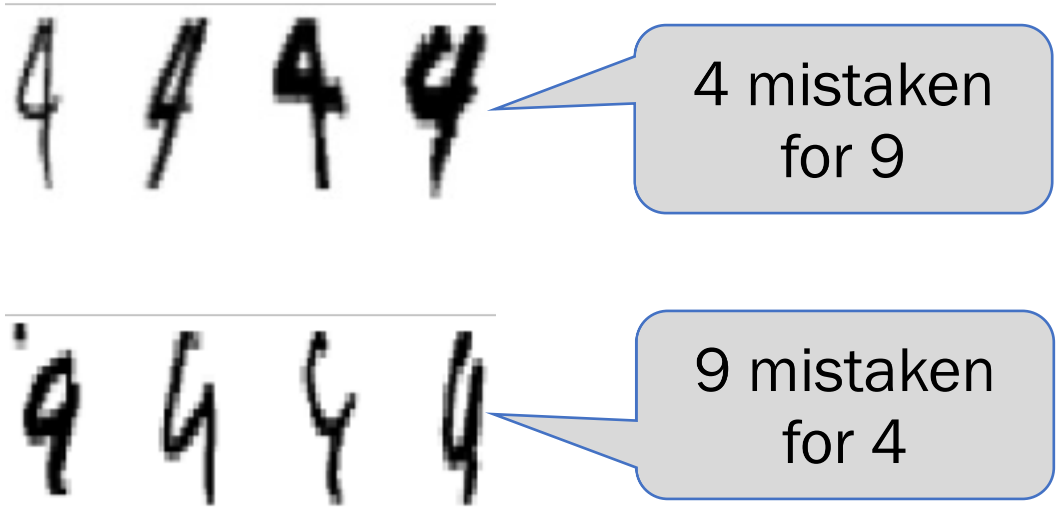
For new sample X_0 compare $\omega^{-1}\big(\mathsf{D}(X_0,\vartheta_{\mathsf{o}})\big)$ to threshold λ to decide whether X_0 statistically follows the first or the second set

Example: Classification Problem

From dataset MNIST isolate handwritten numerals 4 and 9

Training set: 5500 “4” and 5500 “9”. Testing set: 982 “4” and 1009 “9”





Generative Models

X: $\cdots$  $\cdots$ f(X)

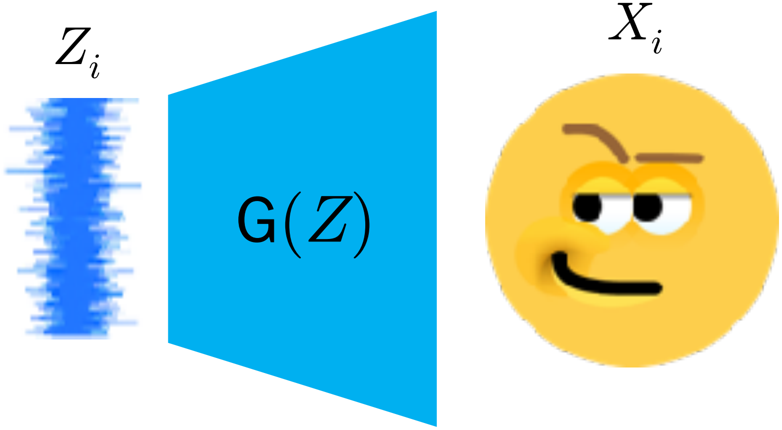
Is it possible to generate synthetic data (realizations X_i) that follow f(X) ?

NOT an easy problem even if density f(X) is known!

Begin with density h(Z): Simple to generate realizations Z_i

Find transformation G(Z): Such that $X_i =$ G(Z_i) follows f(X)

THEOREM: Under general conditions
a transformation G exists !!!



Pair $\{$G(Z),h$(Z)\}$ Generative model

G(Z) Generator

X follows $\mathsf{f}(X)$ and Y follows $\mathsf{g}(Y)$

$$\max_{\mathsf{D}} \mathsf{J}(\mathsf{D}) = \max_{\mathsf{D}} \left\{ \mathbb{E}_{\mathsf{f}}[\phi(\mathsf{D}(X))] + \mathbb{E}_{\mathsf{g}}[\psi(\mathsf{D}(Y))] \right\} \Rightarrow \mathsf{D}_{\mathsf{o}}(X) = \omega\left(\frac{\mathsf{g}(X)}{\mathsf{f}(X)}\right)$$

Z follows $\mathsf{h}(Z)$, select $\mathsf{G}(Z)$, define $Y = \mathsf{G}(Z)$, check if likelihood ratio = 1

THEOREM (Goodfellow et al. 2014): Z follows $\mathsf{h}(Z)$, define $Y = \mathsf{G}(Z)$ and cost

$$\mathsf{J}(\mathsf{G},\mathsf{D}) = \mathbb{E}_{\mathsf{f}}[\phi(\mathsf{D}(X))] + \mathbb{E}_{\mathsf{h}}\Big[\psi\Big(\mathsf{D}(\mathsf{G}(Z))\Big)\Big]$$

then the optimum solution to the adversarial problem

$$\min_{\mathsf{G}} \max_{\mathsf{D}} \mathsf{J}(\mathsf{G},\mathsf{D}) = \min_{\mathsf{G}} \max_{\mathsf{D}} \left\{ \mathbb{E}_{\mathsf{f}}[\phi(\mathsf{D}(X))] + \mathbb{E}_{\mathsf{h}}\Big[\psi\Big(\mathsf{D}(\mathsf{G}(Z))\Big)\Big] \right\}$$

is such that $Y = \mathsf{G}_{\mathsf{o}}(Z)$ follows $\mathsf{f}(Y)$

$\mathsf{D}(X)$ Discriminator $\mathsf{G}(Z)$ Generator

Data Driven Implementation

$\{X_1, X_2, \dots, X_n\}$ following f(X), $\{Z_1, Z_2, \dots, Z_m\}$ following h(Z)

Likelihood ratio D(X) approximated by neural network D(X,ϑ) (Discriminator)

Generator function G(Z) approximated by neural network G(Z,θ) (Generator)

$$\mathsf{J}(\theta,\vartheta)=\frac{1}{n}\sum_{i=1}^{n}\phi\big(\mathsf{D}(X_i,\vartheta)\big)+\frac{1}{m}\sum_{j=1}^{m}\psi\Big(\mathsf{D}\big(\mathsf{G}(Z_j,\theta),\vartheta\big)\Big)$$

Adversarial optimization becomes

$$\min_{\theta}\max_{\vartheta}\mathsf{J}(\theta,\vartheta)=\min_{\theta}\max_{\vartheta}\left\{\frac{1}{n}\sum_{i=1}^{n}\phi\big(\mathsf{D}(X_i,\vartheta)\big)+\frac{1}{m}\sum_{j=1}^{m}\psi\Big(\mathsf{D}\big(\mathsf{G}(Z_j,\theta),\vartheta\big)\Big)\right\}$$

$$\Rightarrow\{\theta_o,\vartheta_o\}\Rightarrow\theta_o\Rightarrow\mathsf{G}(Z,\theta_o)$$

IF Z follows h(Z) THEN $Y=$ G(Z,θ_o) follows f(Y)

Generative
Adversarial
Networks

Example (NVIDIA)

HD-CelebA (30 000 high definition images 1024 X 1024 of celebrities)

NVIDIA used progressive growing of GANs (4X4), (8X8),…,(1024X1024)

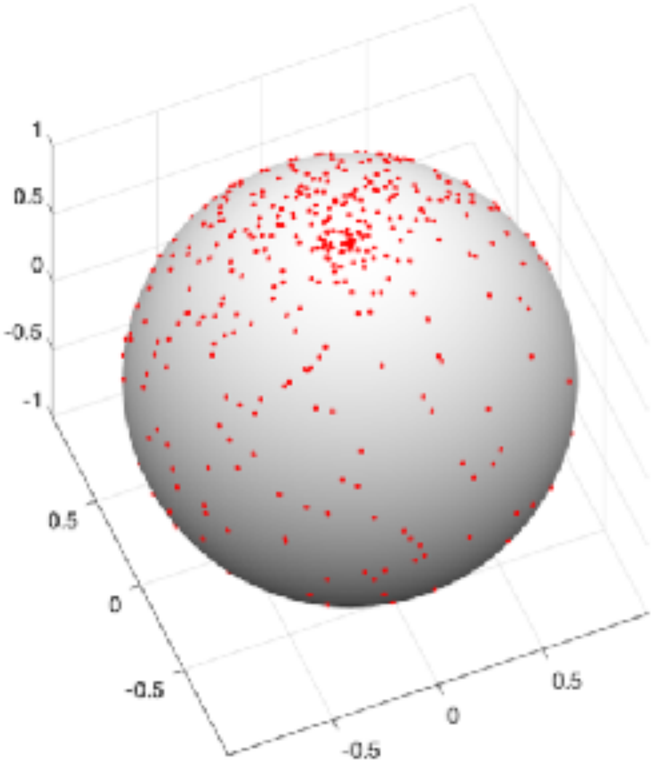
Y of size 3×10^6, Z Gaussian vector of length 500

Generative Models

vs

Probability Densities

Points in N-D space can be random and lie on a lower dimensional surface (manifold)



Example red points on sphere (2-D in 3-D space)

Points are random with coordinates $Y = [y_1, y_2, y_3]$ satisfying the deterministic equation

$$y_1^2 + y_2^2 + y_3^2 = r^2$$

Then density has the form

$$\mathsf{f}(y_1, y_2, y_3) = \delta(y_1^2 + y_2^2 + y_3^2 - r^2)\mathsf{h}(y_1, y_2)$$

Dirac $\delta(x)$ generalized function is defined as

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases}, \quad \int_{-\epsilon}^{\epsilon} \delta(x)\, dx = 1$$

Generative model would describe the random data with input density h(z_1,z_2) and generator vector function G(z_1,z_2)

$$Y = \mathsf{G}(z_1,z_2) \Rightarrow \begin{bmatrix} y_1 = \mathsf{G}_1(z_1,z_2) \\ y_2 = \mathsf{G}_2(z_1,z_2) \\ y_3 = \mathsf{G}_3(z_1,z_2) \end{bmatrix} \Rightarrow \begin{bmatrix} y_1 = r\cos(2\pi z_1)\sin(\pi z_2) \\ y_2 = r\sin(2\pi z_1)\sin(\pi z_2) \\ y_3 = r\cos(\pi z_2) \end{bmatrix}$$

h(z_1,z_2) defined on $[0,1]\times[0,1]$ and G(z_1,z_2) is an ordinary function

Data are representable as $Y=\mathsf{G}(Z)$, Z follows h(Z). Many datasets satisfy

$$\mathsf{dim}(Z) \ll \mathsf{dim}(Y)$$

In HD CelebA: dim$(Y) =$ 3 X 1024 X 1024 = 3 X 10^6
Input to Generator G(Z): dim$(Z) =$ 500 (independent Gaussians)

Application to Inverse Problems

Several image restoration problems in Computer Vision can be formulated as follows

Ideal

Measurement $X = \mathsf{T}(Y) + W$

Known transformation

Noise

more general $X = \mathsf{T}(Y, \alpha) + W$

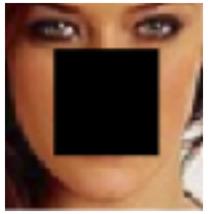
Unknown parameters

Problem: Recover (restore) ideal Y from measurements X

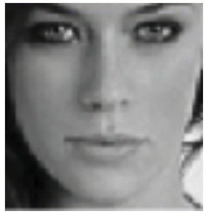
Ideal

Restore

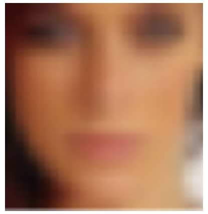
Examples of transformations



Inpainting



Colorization



Super-Res

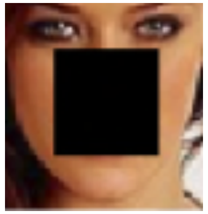
De-Noising

De-Quantization

Recovering Y from measurements X is an ill posed problem

Y

X



Inpainting

More unknowns than equations

Classical approach: Impose "smoothness" constraints to obtain a (unique) solution

…  …

Available generative model $\{\mathsf{G}(Z),\mathsf{h}(Z)\}$: $Y = \mathsf{G}(Z)$

Since $Y = \mathsf{G}(Z)$, instead of estimating Y, estimate input to generator Z then recover Y as the output of the generator

Because $\mathsf{dim}(Z) \ll \mathsf{dim}(Y)$, significant computational gain and stable processing

Ad-Hoc Approaches

Select Z so that measurement X and $\mathsf{T}(\mathsf{G}(Z))$ are "close"

$$\min_{Z} \|X - \mathsf{T}(\mathsf{G}(Z))\|^2 \;\Rightarrow\; Z_o \;\Rightarrow\; Y_o = \mathsf{G}(Z_o)$$

Well defined optimization, computationally stable

Generative model is a pair $\{\mathsf{G}(Z), \mathsf{h}(Z)\}$

Even for $\mathsf{T}(\mathsf{G}(Z_o))$ "close" to X, if likelihood $\mathsf{h}(Z_o)$ is very small

then $Y_o = \mathsf{G}(Z_o)$ is a bad solution

Must take into account input density $\mathsf{h}(Z)$

Yeh et al. (2017), (2018)

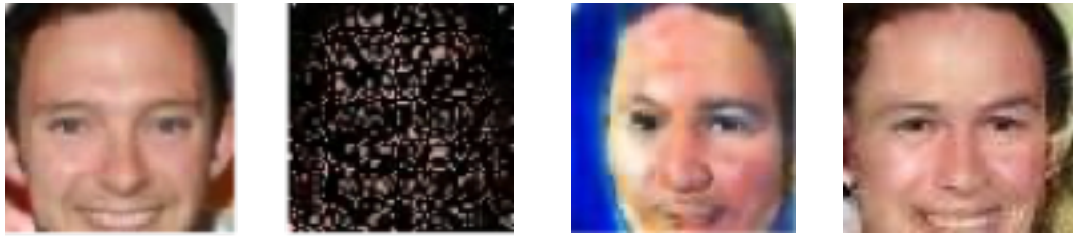
$$\mathsf{J}(Z) = \|X - \mathsf{T}\big(\mathsf{G}(Z)\big)\| + \lambda\left\{ \log\Big(1 - \mathsf{D}\big(\mathsf{G}(Z)\big)\Big) - \log\Big(\mathsf{D}\big(\mathsf{G}(Z)\big)\Big) - \frac{1}{2}\|Z\|^2 \right\}$$

Regularizer

log h(Z)

Parameter needs tuning
Complicated

$$\min_Z \mathsf{J}(Z) \Rightarrow Z_o \Rightarrow Y_o = \mathsf{G}(Z_o)$$



Success ?

Asim et al. (2019)

$$\mathsf{J}(Z) = \|X - \mathsf{T}\big(\mathsf{G}(Z)\big)\|^2 + \lambda\|Z\|^2 \qquad \min_Z \mathsf{J}(Z) \Rightarrow Z_o \Rightarrow Y_o = \mathsf{G}(Z_o)$$

Both methods require exact knowledge of T(Y)

Statistical Estimation

Following classical optimal Statistical estimation theory, in particular the Maximum Aposteriori Probability (MAP) method we obtain

$$\mathsf{J}(Z,\alpha) = \log\left(\|X - \mathsf{T}\big(\mathsf{G}(Z),\alpha\big)\|^2\right) + \frac{1}{N}\|Z\|^2, \quad N = \mathrm{dim}(X)$$

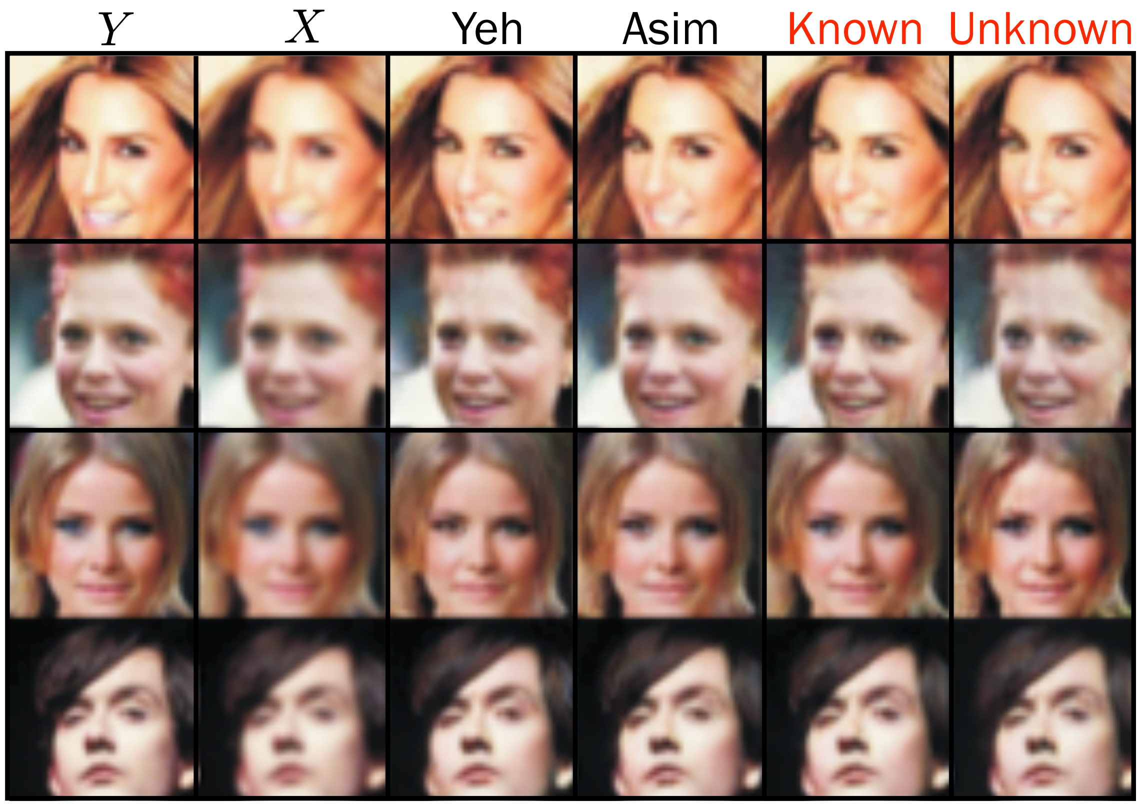
$$\min_{Z,\alpha} \mathsf{J}(Z,\alpha) \;\Rightarrow\; \{Z_\mathsf{o},\alpha_\mathsf{o}\} \;\Rightarrow\; Y_\mathsf{o} = \mathsf{G}(Z_\mathsf{o})$$

No parameters to tune

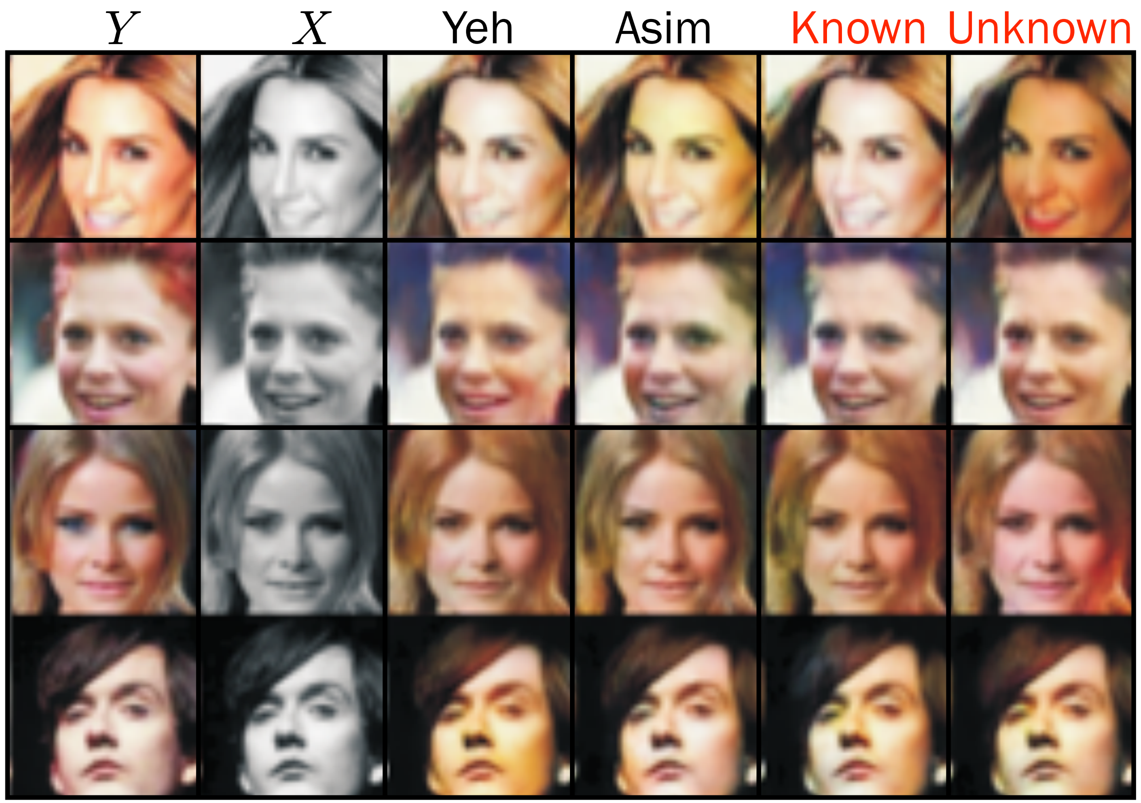
Can accommodate unknown parameters in transformation $\mathsf{T}(Y,\alpha)$

Examples

Blurring with 3 X 3 mask

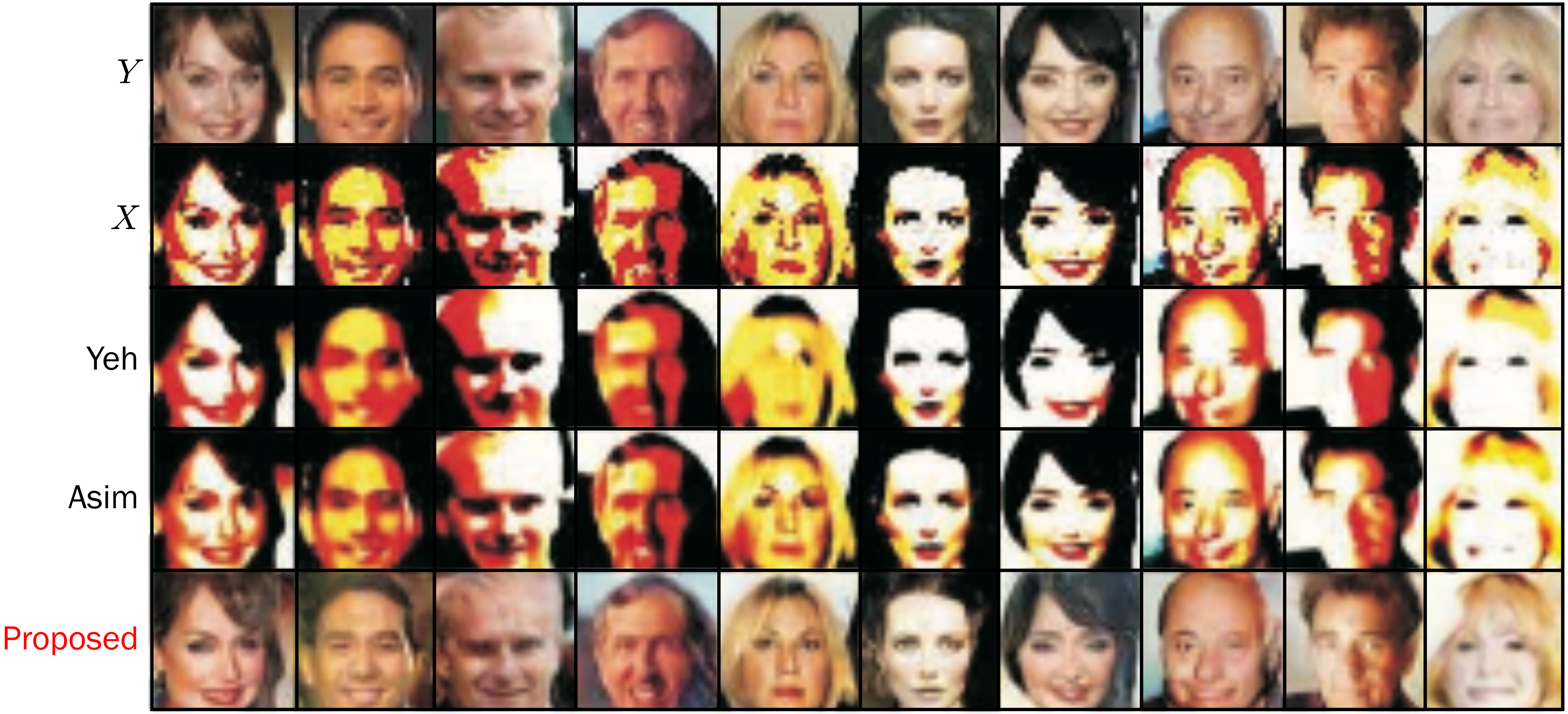


Colorization (green channel)



De-Quantization

2 levels per RGB channel, 8 colors



De-Quantization

3 levels per RGB channel, 27 colors

Y	
X	
Yeh	
Asim	
Proposed	

De-Quantization and Colorization

RGB → Gray → BW (2 levels)

Data Mixtures

$$X = aY + a'Y'$$