From Statistical Estimation and Generative Modeling to Inverse Problems

LECTURE 4: GENERATIVE MODELING

Generative Modeling

- Realizations of random variables
- Classical Methods
  - Inverse cdf
  - Acceptance-rejection
- Using transformations
  - Generative models
  - Design with adversarial methods
  - Design with non-adversarial methods
- Probability density vs Generative model

Realizations of Random Variables

Realizations of random variables are needed in Monte-Carlo simulations

Generate synthetic data (images, music, videos)

When densities of interest are usual then classical methods

In case of modern datasets classical methods fail miserably in producing realizations

Need alternative techniques that can handle data that are multi-dimensional and can lie in lower dimensional manifolds

Regarding random variables that lie on lower dimensional manifolds description using probability densities is not efficient

Classical Methods

For given density $f(x)$ we like to generate realizations

Assume we have probability density $h(z)$ for which we can easily generate realizations, for example Uniform or Gaussian

<u>Inverse CDF</u>

If $z_1, z_2, \ldots$ independent realizations, uniform in [0,1], apply transformation

$$x_i = F^{-1}(z_i)$$

where $F(x) = \int_{-\infty}^{x} f(w)\, dw$ is the cumulative distribution function (cdf)

and $F^{-1}(z)$ its inverse function ($F^{-1}(z) = x \Leftrightarrow z = F(x)$)

x_i follows $f(x)$

$$\mathbb{P}(x \le \mathsf{x}) = \mathbb{P}\big(\mathsf{F}^{-1}(z) \le \mathsf{x}\big) = \mathbb{P}\big(z \le \mathsf{F}(\mathsf{x})\big) = \mathsf{F}(\mathsf{x})$$

Can we extend this to multi-dimensional densities?

If we have joint density f(x_1,x_2,x_3) can we generate triplets?

Bayes Rule: $\mathsf{f}(x_1, x_2, x_3) = \mathsf{f}(x_3|x_2, x_1)\mathsf{f}(x_2|x_1)\mathsf{f}(x_1)$

Conditionally independent

Generate independent uniform realizations $z_1,\ z_2,\ z_3$

$$\mathsf{f}(x_1) \ \Rightarrow\ \mathsf{F}(x_1) \ \Rightarrow\ x_1 = \mathsf{F}^{-1}(z_1)$$

$$\mathsf{f}(x_2|x_1) \ \Rightarrow\ \mathsf{F}(x_2|x_1) \ \Rightarrow\ x_2 = \mathsf{F}^{-1}(z_2|x_1)$$

$$\mathsf{f}(x_3|x_2, x_1) \ \Rightarrow\ \mathsf{F}(x_3|x_2, x_1) \ \Rightarrow\ x_3 = \mathsf{F}^{-1}(z_3|x_2, x_1)$$

then x_1,x_2,x_3 follow f(x_1,x_2,x_3)

Not practically convenient !!!

Acceptance/Rejection

We are given f(x), cannot compute F(x) or F$^{-1}(z)$, can evaluate f(x)

Assume another density h(z) for which we can generate realizations (e.g. Gaussian)

Assume we know L such that $\frac{f(x)}{h(x)} \leq L < \infty$ for all x

Generate pair (z_i, t_i), independent with $z_i \sim h(z)$ and $t_i \sim U([0,1])$

If $\frac{f(z_i)}{h(z_i)} \geq L\, t_i$ then accept and set $x_i = z_i$

Otherwise reject pair and try again with a new one

x_i follows f(x)

True for vector densities f(X) when can generate h(Z)

Using Transformations

Generative Models

Want realizations X_i that follow f(X). Common method using transformations

Start with $Z \sim$ h(Z), find G(Z) deterministic so that

$X =$ G(Z) follows f(X)

Does G(Z) exist ?

THEOREM: Under general conditions

YES IT EXISTS !!!

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

Theorem proves existence

Scalar $z \sim \mathsf{h}(z)$, strictly increasing $\mathsf{G}(z)$, then density of $x = \mathsf{G}(z)$ can be found

$$\mathbb{P}(x \leq \mathsf{x}) = \mathbb{P}(\mathsf{G}(z) \leq \mathsf{x}) = \mathbb{P}\big(z \leq \mathsf{G}^{-1}(\mathsf{x})\big) = \mathsf{H}\big(\mathsf{G}^{-1}(\mathsf{x})\big)$$

$\mathsf{z}(\mathsf{x}) = \mathsf{G}^{-1}(\mathsf{x})$ is the unique solution to $\mathsf{G}(\mathsf{z}) = \mathsf{x}$

$$\Rightarrow \; \mathsf{f}(x) = \mathsf{h}\big(\mathsf{G}^{-1}(x)\big) \times \frac{1}{\mathsf{G}'\big(\mathsf{G}^{-1}(x)\big)} = \mathsf{h}\big(\mathsf{z}(x)\big) \times \frac{1}{|\mathsf{G}'\big(\mathsf{z}(x)\big)|}$$

If $\mathsf{G}(z)$ not monotone then $\mathsf{G}(\mathsf{z}) = \mathsf{x}$ may have multiple solutions: $\mathsf{z}_1(\mathsf{x}), \ldots, \mathsf{z}_p(\mathsf{x})$

$$\mathsf{f}(x) = \mathsf{h}\big(\mathsf{z}_1(x)\big) \times \frac{1}{|\mathsf{G}'\big(\mathsf{z}_1(x)\big)|} + \cdots + \mathsf{h}\big(\mathsf{z}_p(x)\big) \times \frac{1}{|\mathsf{G}'\big(\mathsf{z}_p(x)\big)|}$$

Vector $Z \sim \mathsf{h}(Z)$ and vector transformation $\mathsf{G}(Z)$

Random vector $X = \mathsf{G}(Z)$, how to compute density?

Equation $\mathsf{G}(\mathsf{Z}) = \mathsf{X}$ has multiple solutions $\mathsf{Z}_1(\mathsf{X}), \ldots, \mathsf{Z}_p(\mathsf{X})$

$$\mathsf{f}(X) = \frac{\mathsf{h}\big(\mathsf{Z}_1(X)\big)}{\left|\det\left\{\mathbb{J}_Z \mathsf{G}\big(\mathsf{Z}_1(X)\big)\right\}\right|} + \cdots + \frac{\mathsf{h}\big(\mathsf{Z}_p(X)\big)}{\left|\det\left\{\mathbb{J}_Z \mathsf{G}\big(\mathsf{Z}_p(X)\big)\right\}\right|}$$

Target

$$\mathsf{G}(Z) = \begin{bmatrix} \mathsf{g}_1(z_1, \ldots, z_k) \\ \vdots \\ \mathsf{g}_k(z_1, \ldots, z_k) \end{bmatrix}, \quad \mathbb{J}_Z \mathsf{G}(Z) = \begin{bmatrix} \frac{\partial \mathsf{g}_1}{\partial z_1} & \cdots & \frac{\partial \mathsf{g}_1}{\partial z_k} \\ \vdots & \cdots & \vdots \\ \frac{\partial \mathsf{g}_k}{\partial z_1} & \cdots & \frac{\partial \mathsf{g}_k}{\partial z_k} \end{bmatrix}$$

To design

Why?? To generate realizations of X following $\mathsf{f}(X)$ by generating realizations of Z following $\mathsf{h}(Z)$ and then transforming $X = \mathsf{G}(Z)$

Adversarial Methods (GANs)

Goal: Start with Z following h(Z). Find generative transformation G(Z) so that X=G(Z) follows f(X). Solve problem without knowing f(X) !!!

Let us develop our method following a step-by-step procedure

Suppose we have two probability densities f(X) and g(X)
How can we test f$(X) =$ g(X) ?

$$\text{Sufficient:} \quad \frac{\text{g}(X)}{\text{f}(X)} = 1 \quad \equiv \quad \omega\left(\frac{\text{g}(X)}{\text{f}(X)}\right) = \omega(1)$$

for ANY $\omega(\cdot)$ strictly increasing

Suppose we do not have f(X), g(X) but still desire to test f$(X) =$ g(X)

For every function $\delta(X)$ mechanism provides $\mathbb{E}_{\text{f}}[\delta(X)], \;\; \mathbb{E}_{\text{g}}[\delta(X)]$

Can we compute $\mathrm{D}(X) = \omega\left(\frac{\mathrm{g}(X)}{\mathrm{f}(X)}\right)$ using averages ? YES !!!

THEOREM:

Specify strictly increasing function $\omega(\mathrm{r})$ and positive function $\rho(z)$

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

and for any scalar function D(X) the cost

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

Then the optimum solution to the maximization problem $\max_{\mathrm{D}(X)} \mathrm{J(D)}$ satisfies

$$\mathrm{D}_{\mathrm{o}}(X) = \omega\left(\frac{\mathrm{g}(X)}{\mathrm{f}(X)}\right)$$

Why insist on averages ?

If we do not have $\mathsf{f}(X)$, $\mathsf{g}(Y)$ but two sets of realizations

$$X_1, X_2, \ldots, X_n \quad \sim \quad \mathsf{f}(X) \;\rightarrow\; \mathbb{E}_{\mathsf{f}}\big[\delta(X)\big] \approx \frac{1}{n}\sum_{t=1}^{n}\delta(X_t)$$

$$Y_1, Y_2, \ldots, Y_m \quad \sim \quad \mathsf{g}(Y) \;\rightarrow\; \mathbb{E}_{\mathsf{g}}\big[\delta(Y)\big] \approx \frac{1}{m}\sum_{t=1}^{m}\delta(Y_t)$$

How can we optimize with respect to unknown function $\mathsf{D}(X)$?

Replace $\mathsf{D}(X)$ with neural network $\mathsf{D}(X,\vartheta)$. Optimize over network parameters ϑ

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

$$\hat{\mathsf{J}}(\vartheta) = \frac{1}{n}\sum_{t=1}^{n}\phi\big(\mathsf{D}(X_t,\vartheta)\big) + \frac{1}{m}\sum_{t=1}^{m}\psi\big(\mathsf{D}(Y_t,\vartheta)\big)$$

Solve $\max_{\vartheta} \hat{\mathrm{J}}(\vartheta) \Rightarrow \vartheta_{\mathrm{o}} \Rightarrow \mathrm{D}(X, \vartheta_{\mathrm{o}})$

Expect $\mathrm{D}(X, \vartheta_{\mathrm{o}}) \approx \mathrm{D}_{\mathrm{o}}(X) = \omega\left(\frac{\mathrm{g}(X)}{\mathrm{f}(X)}\right)$

Can we use function $\mathrm{D}(X,\vartheta_{\mathrm{o}})$ to examine the two densities $\mathrm{f}(X)$, $\mathrm{g}(Y)$?

Employ $\mathrm{D}(X,\vartheta_{\mathrm{o}})$ to discriminate between $\mathrm{f}(X)$, $\mathrm{g}(Y)$ using only data

$$X_1, X_2, \ldots, X_n \sim \mathrm{f}(X) \qquad Y_1, Y_2, \ldots, Y_m \sim \mathrm{g}(Y)$$

If $\mathrm{D}(X, \vartheta_{\mathrm{o}}) \not\approx \omega(1) \Rightarrow$ the two datasets have different densities

If $\mathrm{D}(X, \vartheta_{\mathrm{o}}) \approx \omega(1) \Rightarrow$ the two datasets have similar densities

$\mathrm{D}(X,\vartheta)$ is known as the Discriminator function

Examples

A: $\omega(\mathrm{r}) = \mathrm{r}$

$$\rho(z) = 1, \quad \Rightarrow \quad \phi(z) = -\frac{z^2}{2}, \quad \psi(z) = z$$

(Mean Square)

B: $\omega(\mathrm{r}) = \log(\mathrm{r})$

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

(Exponential)

C: $\omega(\mathrm{r}) = \frac{\mathrm{r}}{\mathrm{r}+1}$ Goodfellow et al. (2014), NeurIPS

$$\rho(z) = \frac{1}{z}, \quad \Rightarrow \quad \phi(z) = \log(1-z), \quad \psi(z) = \log(z)$$

(Cross Entropy)

What happened to the Generator ?

Generator $\mathsf{G}(Z)$ transforms Z to $Y = \mathsf{G}(Z)$ with Z ~ $\mathsf{h}(Z)$

We desire Y ~ f(.) same density as training data $\{X_1,\ldots,X_n\}$

Naïve Method: Select a $\mathsf{G}(Z)$. Test if transformation is the desired
If not, make another selection

How do we test if selection is any good ?

Generate realizations of Z: $\{Z_1,\ldots,Z_m\}$
Apply generator to samples, create $\{Y_1,\ldots,Y_m\}$ where $Y_i = \mathsf{G}(Z_i)$
Test $\{Y_1,\ldots,Y_m\}$ against $\{X_1,\ldots,X_n\}$

$$\hat{\mathsf{J}}(\vartheta) = \frac{1}{n}\sum_{t=1}^{n}\phi\big(\mathsf{D}(X_t,\vartheta)\big) + \frac{1}{m}\sum_{t=1}^{m}\psi\big(\mathsf{D}(Y_t,\vartheta)\big)$$

$$\max_{\vartheta}\hat{\mathsf{J}}(\vartheta) \Rightarrow \vartheta_o \Rightarrow \mathsf{D}(X,\vartheta_o)$$

$$\mathsf{D}(X,\vartheta_o) \approx \omega\left(\frac{\mathsf{g}(X)}{\mathsf{f}(X)}\right)$$

$$\mathsf{D}(X,\vartheta_o) \not\approx \omega(1)$$

$$\hat{\mathsf{J}}(\vartheta) = \frac{1}{n}\sum_{t=1}^{n}\phi\big(\mathsf{D}(X_t,\vartheta)\big) + \frac{1}{m}\sum_{t=1}^{m}\psi\big(\mathsf{D}(Y_t,\vartheta)\big)$$

$$\hat{\mathsf{J}}(\vartheta) = \frac{1}{n}\sum_{t=1}^{n}\phi\big(\mathsf{D}(X_t,\vartheta)\big) + \frac{1}{m}\sum_{t=1}^{m}\psi\Big(\mathsf{D}\big(\mathsf{G}(Z_t),\vartheta\big)\Big)$$

We are looking for function $\mathsf{G}(Z)$. Use neural network $\mathsf{G}(Z,\theta)$

$$\hat{\mathsf{J}}(\theta,\vartheta) = \frac{1}{n}\sum_{t=1}^{n}\phi\big(\mathsf{D}(X_t,\vartheta)\big) + \frac{1}{m}\sum_{t=1}^{m}\psi\Big(\mathsf{D}\big(\mathsf{G}(Z_t,\theta),\vartheta\big)\Big)$$

<u>THEOREM</u>: For fixed θ (generator $\mathsf{G}(Z,\theta)$) we have

$$\max_{\vartheta}\hat{\mathsf{J}}(\theta,\vartheta) \gtrapprox \phi\big(\omega(1)\big) + \psi\big(\omega(1)\big)$$

Equality when $Y = \mathsf{G}(Z,\theta)$ has density f(.) same as $\{X_1,\dots,X_n\}$

Since $\max_{\vartheta} \hat{\mathrm{J}}(\theta, \vartheta) \gtrapprox \phi\big(\omega(1)\big) + \psi\big(\omega(1)\big)$

to bring $\max_{\vartheta} \hat{\mathrm{J}}(\theta, \vartheta)$ as close as possible to $\phi\big(\omega(1)\big) + \psi\big(\omega(1)\big)$

we must apply minimization over θ

$$\min_{\theta} \max_{\vartheta} \hat{\mathrm{J}}(\theta, \vartheta)$$

$$= \min_{\theta} \max_{\vartheta} \left\{ \frac{1}{n} \sum_{t=1}^{n} \phi\big(\mathrm{D}(X_t, \vartheta)\big) + \frac{1}{m} \sum_{t=1}^{m} \psi\Big(\mathrm{D}\big(\mathrm{G}(Z_t, \theta), \vartheta\big)\Big) \right\}$$

Adversarial Optimization → GAN

G Generative
A Adversarial
N Networks

We have Z ~ $\mathsf{h}(Z)$
Design generator $\mathsf{G}(Z)$ so that $Y = \mathsf{G}(Z)$ has the same density as X
for which we have realizations (training set) $\{X_1, \ldots, X_n\}$

Approximate generator with neural network $\mathsf{G}(Z, \theta)$
Define second neural network the discriminator $\mathsf{D}(X, \vartheta)$

For realizations of Z: $\{Z_1, \ldots, Z_m\}$ consider adversarial problem

$$\min_{\theta} \max_{\vartheta} \hat{\mathsf{J}}(\theta, \vartheta)$$
$$= \min_{\theta} \max_{\vartheta} \left\{ \frac{1}{n} \sum_{t=1}^{n} \phi\big(\mathsf{D}(X_t, \vartheta)\big) + \frac{1}{m} \sum_{t=1}^{m} \psi\Big(\mathsf{D}\big(\mathsf{G}(Z_t, \theta), \vartheta\big)\Big) \right\}$$

then generator $\mathsf{G}(Z, \theta_o)$ when applied to realizations of Z yields samples following closely the density of $\{X_1, \ldots, X_n\}$

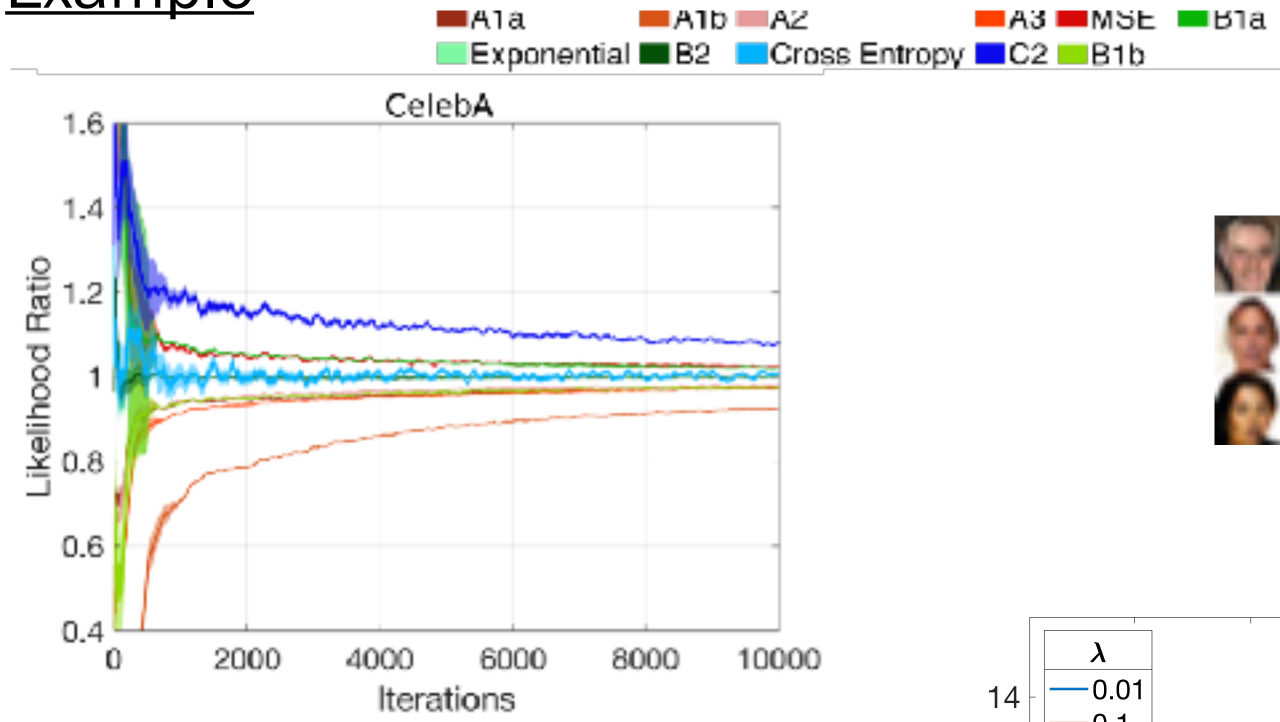
Example

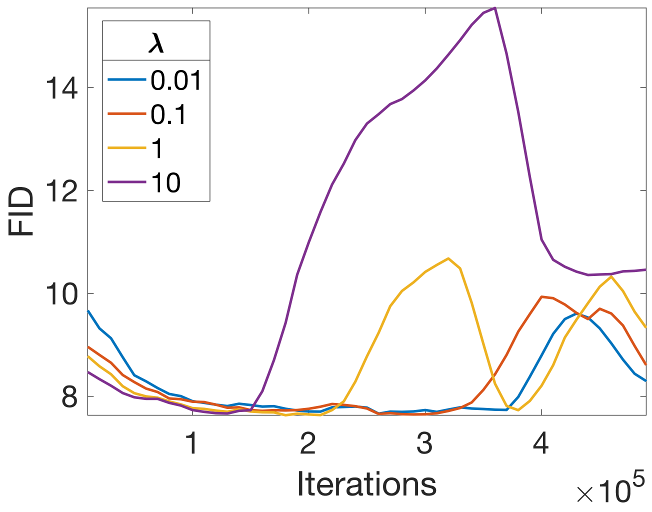
High definition CelebA (30 000 high definition images 1024 X 1024 of celebrities)

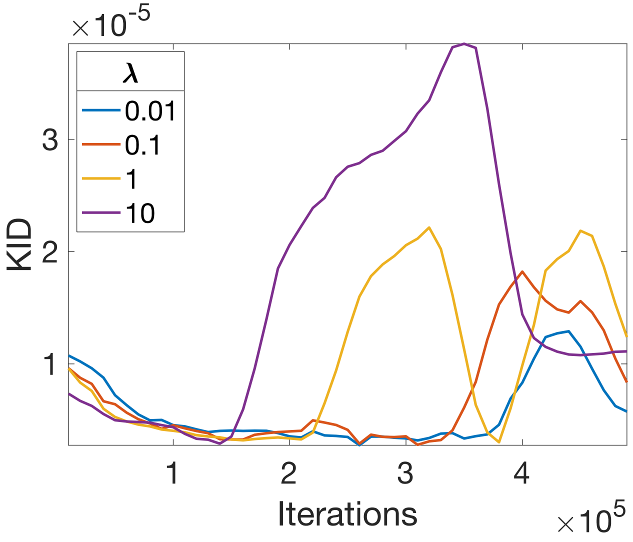
Extremely hard to control convergence of the adversarial problem

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

Example







Convergence/Divergence

Non-Adversarial Method

Interested in Generator
Discriminator used during generator design, afterwards useless

Possible (at least in theory) to design Generator without Discriminator by not employing min-max (adversarial) optimization

We can define optimization criteria involving only maximization or minimization with the help of positive definite kernels and design successfully Generators

A symmetric scalar function $\mathsf{K}(X,Y)$ will be positive definite if for every nonzero function $\varphi(X)$ it satisfies

$$\iint \mathsf{K}(X,Y)\varphi(X)\varphi(Y)\,dX\,dY \;>\; 0$$

Gaussian Kernel: $\mathsf{K}(X,Y) = e^{-\frac{1}{h^2}\|X-Y\|^2}$

Validity of the Cauchy-Schwarz inequality

$$\left(\iint \mathsf{K}(X,Y)\phi(X)\psi(Y)\,dX\,dY\right)^2$$
$$\leq \left(\iint \mathsf{K}(X,Y)\phi(X)\phi(Y)\,dX\,dY\right)\left(\iint \mathsf{K}(X,Y)\psi(X)\psi(Y)\,dX\,dY\right)$$

For densities $\mathsf{f}(X)$, $\mathsf{g}(Y)$

$$\frac{\left(\iint \mathsf{K}(X,Y)\mathsf{f}(X)\mathsf{g}(Y)\,dX\,dY\right)^2}{\iint \mathsf{K}(X,Y)\mathsf{f}(X)\mathsf{f}(Y)\,dX\,dY} \leq \iint \mathsf{K}(X,Y)\mathsf{g}(X)\mathsf{g}(Y)\,dX\,dY$$

with equality if and only if $\mathsf{f}(X) = \mathsf{g}(Y)$

$$\frac{\Big(\iint \mathsf{K}(X,Y)\mathsf{f}(X)\mathsf{g}(Y)\,dX\,dY\Big)^2}{\iint \mathsf{K}(X,Y)\mathsf{f}(X)\mathsf{f}(Y)\,dX\,dY} \leq \iint \mathsf{K}(X,Y)\mathsf{g}(X)\mathsf{g}(Y)\,dX\,dY$$

$$\frac{\Big(\mathbb{E}\big[\mathsf{K}(X,Y)\big]\Big)^2}{\mathbb{E}\big[\mathsf{K}(Y^1,Y^2)\big]} \leq \mathbb{E}\big[\mathsf{K}(X^1,X^2)\big]$$

where X^1, X^2 independent with the same density $\mathsf{f}(X)$
and Y^1, Y^2 independent with the same density $\mathsf{g}(Y)$

$$\mathsf{J}(\theta) = \frac{\Big(\mathbb{E}\big[\mathsf{K}\big(X,\mathsf{G}(Z,\theta)\big)\big]\Big)^2}{\mathbb{E}\big[\mathsf{K}\big(\mathsf{G}(Z^1,\theta),\mathsf{G}(Z^2,\theta)\big)\big]} \leq \mathbb{E}\big[\mathsf{K}(X^1,X^2)\big]$$

where Z^1, Z^2 independent and follow $\mathsf{h}(Z)$

$$\max_{\theta}\ \mathsf{J}(\theta)$$

We have $Z \sim \mathsf{h}(Z)$

Design generator $\mathsf{G}(Z)$ so that $Y = \mathsf{G}(Z)$ has the same density as X for which we have realizations (training set) $\{X_1,\ldots,X_n\}$

Generate $\{Z_1,\ldots,Z_m\}$ from $\mathsf{h}(Z)$

$$\hat{\mathsf{J}}(\theta) = \frac{\left(\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathsf{K}\big(X_i,\mathsf{G}(Z_j,\theta)\big)\right)^2}{\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j=1,j\neq i}^{m}\mathsf{K}\big(\mathsf{G}(Z_i,\theta),\mathsf{G}(Z_j,\theta)\big)}$$

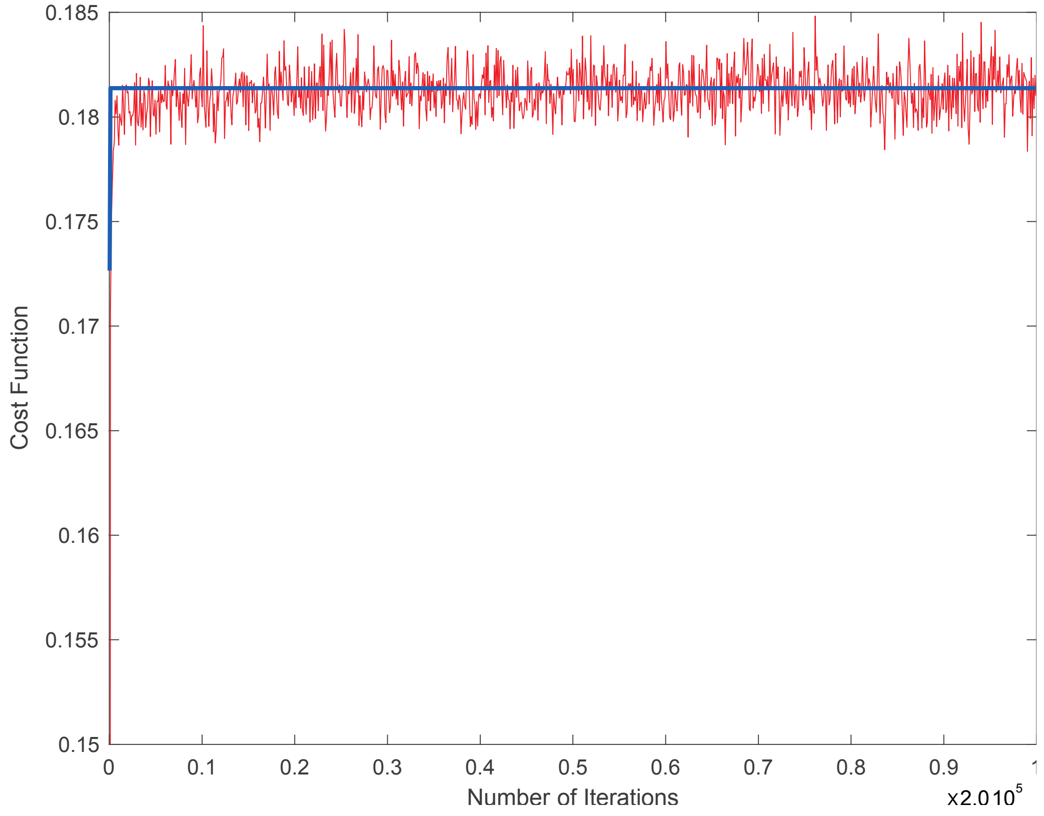
$$\lessapprox \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\mathsf{K}(X_i,X_j)$$

$$\max_{\theta}\hat{\mathsf{J}}(\theta)$$

No Convergence/Divergence phenomenon!!!!

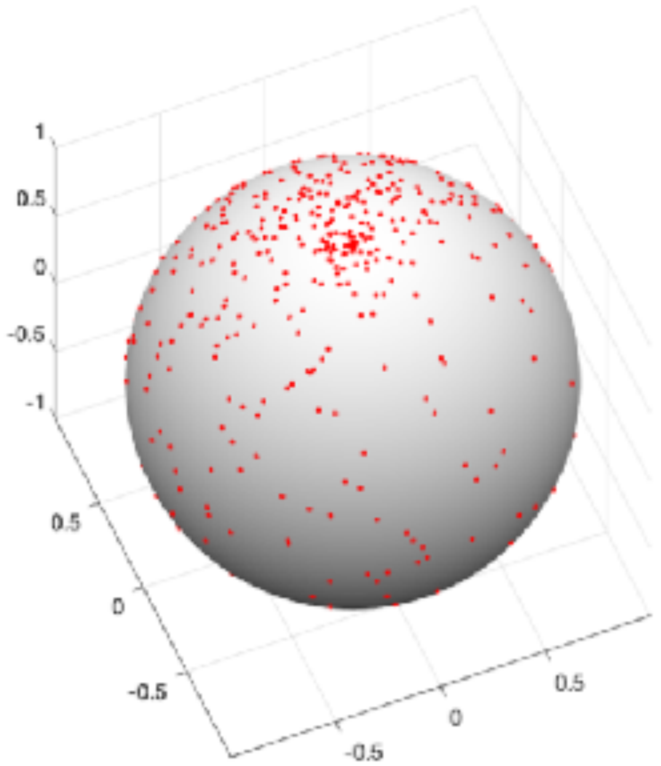
Example

CelebA images 32 X 32



Suitable for small sized and fully connected networks

Probability Density vs Generative Model



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 $[x_1,\ x_2,\ x_3]$ related through a deterministic equation

To lie on a sphere of radius r : $x_1^2 + x_2^2 + x_3^2 = r^2$

$$\mathrm{f}(x_1, x_2, x_3) = \delta(x_1^2 + x_2^2 + x_3^2 - r^2)\, \mathrm{h}(x_1, x_2)$$

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 $\mathsf{h}(z_1,z_2)$ and generator vector function $\mathsf{G}(z_1,z_2)$

$$X = \mathsf{G}(z_1, z_2) \Rightarrow \begin{bmatrix} x_1 = \mathsf{G}_1(z_1, z_2) \\ x_2 = \mathsf{G}_2(z_1, z_2) \\ x_3 = \mathsf{G}_3(z_1, z_2) \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 = r \cos(2\pi z_1) \sin(\pi z_2) \\ x_2 = r \sin(2\pi z_1) \sin(\pi z_2) \\ x_3 = r \cos(\pi z_2) \end{bmatrix}$$

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

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

$$\dim(Z) \ll \dim(X)$$

In HD CelebA: $\dim(X) = 3 \times 1024 \times 1024 = 3 \times 10^6$
Input to Generator $\mathsf{G}(Z)$: $\dim(Z) = 500$ (independent Gaussians)

Instead of estimating X, we first estimate Z, then recover X as $X = \mathsf{G}(Z)$