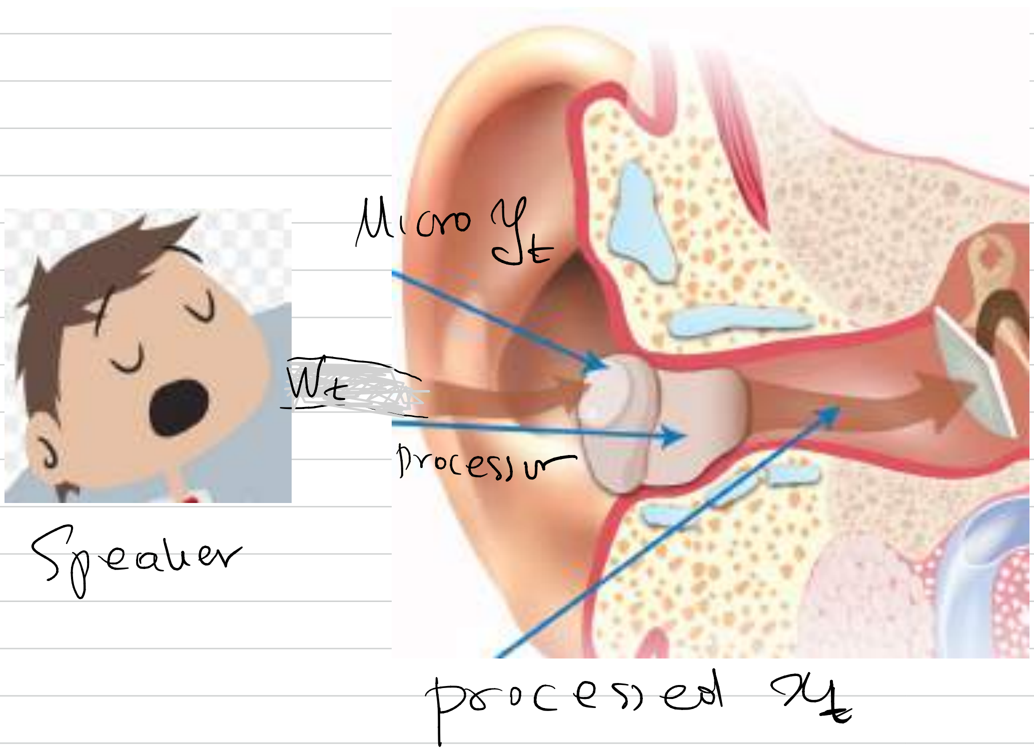
Data-Driven
Bayesian and Non-Bayesian
Parameter Estimation

After detecting an airplane we would like to estimate its position and speed using the radar data $\{x_1, \ldots, x_N\}$



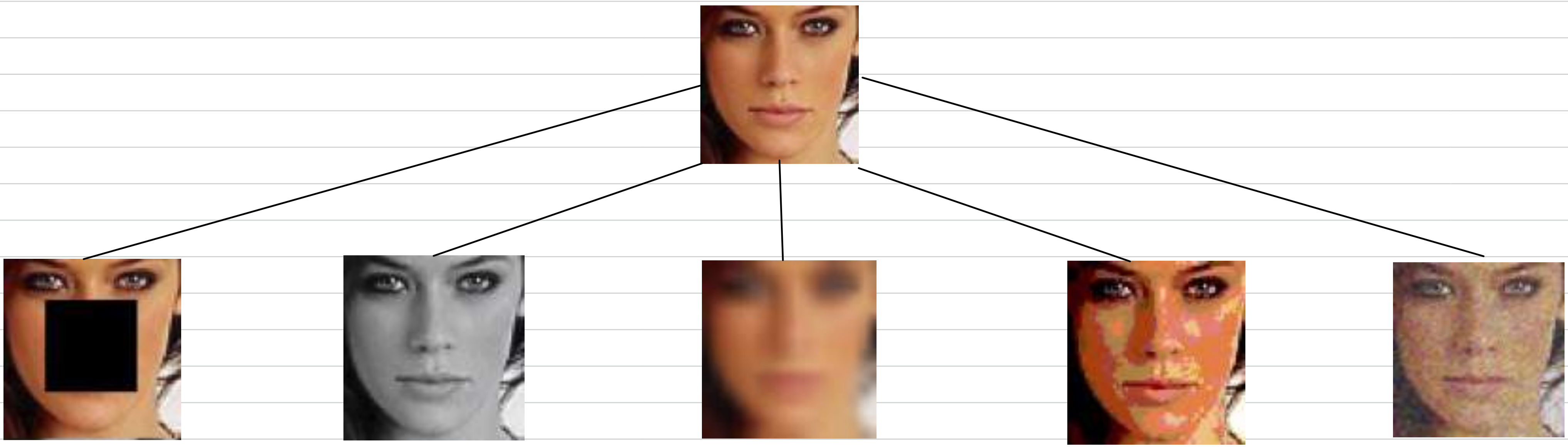
$$y_t = w_t + \underbrace{h_1 x_{t-\tau_1} + \cdots + h_k x_{t-\tau_k}}_{\text{echo cancelation.}}$$

(removal)

In many problems we would like to measure X but instead we measure Y. Can we recover (estimate) X by processing Y?

X plays the role of "parameters"

Image Restoration (inverse problems)



Inpainting Colorization Super-Resolution De-Quantization De-Noising

Classical Parameter Estimation

Bayesian Approach

Data measurements $X = \{x_1, \ldots, x_N\}$

X is realization of random vector $\mathcal{X}$ with probability density $\mathsf{f}(X|\theta)$ which known up to parameter vector θ

What of θ?

θ is a realization of a random vector ϑ with density $\mathsf{p}(\theta)$. It expresses the prior knowledge about the parameters from observations made in past

Random vectors $\mathcal{X}, \vartheta$ are statistically related and relationship captured by the joint density

$$\mathsf{f}(X, \theta) = \mathsf{f}(X|\theta)\mathsf{p}(\theta)$$

Problem of Interest

"Nature" selects a realization θ of ϑ that follows density $\mathsf{p}(\theta)$

For the given θ "Nature" generates a realization X of $\mathcal{X}$ that follows $\mathsf{f}(X|\theta)$

From the measurements X and the prior information $\mathsf{f}(X|\theta)$, $\mathsf{p}(\theta)$
Estimate the θ that gave rise to the measurements

What is an "Estimator"??

Assume $\{x_1, \ldots, x_N\}$ realizations of a random variable χ with mean μ. We want to estimate μ

$$\hat{\mu}_1 = \frac{x_1 + \cdots + x_N}{N} \qquad \hat{\mu}_2 = \frac{e^{x_1} + \cdots + e^{x_N}}{N^2}$$

ANY function of the measurements can play the role of an estimator!!

Performance Computation

Define a cost function $\mathsf{C}(\hat{\theta}, \theta)$

Compute Average Cost

$$\mathcal{C}(\hat{\theta}) = \mathbb{E}_{\mathcal{X},\vartheta}\left[\mathsf{C}\left(\hat{\theta}(\mathcal{X}), \vartheta\right)\right]$$

Minimize Average Cost with respect to function $\hat{\theta}(X)$

$$G(U, X) = \int \mathsf{C}(U, \theta)\mathsf{f}(\theta|X)d\theta = \frac{\int \mathsf{C}(U, \theta)\mathsf{f}(X|\theta)\mathsf{p}(\theta)d\theta}{\int \mathsf{f}(X|\theta)\mathsf{p}(\theta)d\theta}$$

$$\boxed{\hat{\theta}_o(X) = \arg\min_U G(U, X)}$$

Examples

$C(U,\theta) = \|U - \theta\|^2$ MMSE

Minimum Mean Square Error

$$\hat{\theta}_{\text{MMSE}}(X) = \mathbb{E}[\vartheta|X] = \int \theta \mathsf{f}(\theta|X)d\theta = \frac{\int \theta \mathsf{f}(X|\theta)\mathsf{p}(\theta)d\theta}{\int \mathsf{f}(X|\theta)\mathsf{p}(\theta)d\theta}$$

$C(U,\theta) = \|U - \theta\|_{L_1} = |U_1 - \theta_1| + \cdots + |U_k - \theta_k|$ MAE

Minimum Mean Absolute Error

for scalar U, θ

$$\hat{\theta}_{\text{MAE}}(X) = \arg\left\{U : \int_{-\infty}^{U} \mathsf{f}(\theta|X)d\theta = \frac{1}{2}\right\}$$

$$C(U,\theta) = \begin{cases} 1 & \text{when } \|U-\theta\| \geq \delta, \\ 0 & \text{when } \|U-\theta\| < \delta, \end{cases} \quad \delta \to 0$$ MAP

Maximum Aposteriori Probability

$$\hat{\theta}_{\text{MAP}}(X) = \arg\max_{\theta} \mathsf{f}(\theta|X) = \arg\max_{\theta} \mathsf{f}(X|\theta)\mathsf{p}(\theta)$$

Basic Tools

Neural Netwoks

Special class of parametric functions $u(X, \alpha)$ where
α the network parameters

Any function $v(X)$ can be approximated arbitrarily close by a neural network of sufficiently high order

Searching over α in a neural network $u(X, \alpha)$ corresponds to searching over any function $v(X)$ when the size of the network becomes arbitrarily large

Law of Large Numbers (LLN)

$\mathcal{X}$ random and $\{X_1, X_2, \ldots, X_N\}$ realizations

Let $G(X)$ be a deterministic function, then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} G(X_i) = \mathbb{E}_{\mathcal{X}}[G(\mathcal{X})] = \int G(X)\mathsf{f}(X)dX$$

Gradient Descent

Deterministic function $J(\theta)$ and interested in $\min_{\theta} J(\theta)$

We can apply: $\boxed{\theta_t = \theta_{t-1} - \mu \nabla_\theta J(\theta_{t-1})}$

Stochastic Gradient Descent

$$J(\theta) = \mathbb{E}_{\mathcal{X}}[G(\mathcal{X}, \theta)]$$

Instead of $\mathsf{f}(X)$ we have $\{X_1, \ldots, X_N\}$ then

$$\boxed{\theta_t = \theta_{t-1} - \mu \nabla_\theta G(X_t, \theta_{t-1}), \quad \mu > 0}$$

Data - Driven Version

Classical estimation assumes availability of $\mathsf{f}(X|\theta), \mathsf{p}(\theta)$
Equivalent to joint density $\mathsf{f}(X, \theta)$

$\mathsf{f}(X, \theta)$ expresses the random relationship between $\mathcal{X}$ and ϑ

If $\mathsf{f}(X, \theta)$ to be replaced by data then we need pairs

$$\{(X_1, \theta_1), (X_2, \theta_2), \ldots, (X_N, \theta_N)\}$$

The general estimator function $\hat{\theta}(X)$ is replaced by $u(X, \alpha)$
a neural network with parameters α

The optimization becomes

$$\min_{\hat{\theta}(X)} \mathbb{E}_{\mathcal{X},\vartheta}\left[\mathsf{C}\left(\hat{\theta}(\mathcal{X}), \vartheta\right)\right] \Rightarrow \min_{\alpha} \mathbb{E}_{\mathcal{X},\vartheta}\left[\mathsf{C}\left(u(\mathcal{X}, \alpha), \vartheta\right)\right]$$

10

Apply Stochastic Gradient Descent

$$\alpha_t = \alpha_{t-1} - \mu\left[\mathbb{J}_\alpha u(X_t, \alpha_{t-1})\right]^\mathsf{T} \nabla_U \mathsf{C}\big(u(X_t, \alpha_{t-1}), \theta_t\big)$$

Alternatively replace expectation using LLN

$$\min_\alpha \sum_{i=1}^{N} \mathsf{C}\big(u(X_i, \alpha), \theta_i\big)$$

Solve using Gradient Descent with respect to α

If limit is α_o then we expect

$$u(X, \alpha_o) \approx \hat{\theta}_o(X)$$

11

Non-Bayesian Estimation

"Nature" selects θ and generates measurements X that follow $\mathsf{f}(X|\theta)$

We know $\mathsf{f}(X|\theta)$ up to a set of parameters θ

Parameters θ are deterministic and unknown

Problem: Given measurement vector X estimate θ

$$\hat{\theta}_{\mathrm{MLE}}(X) = \arg\max_{\theta} \mathsf{f}(X|\theta)$$

Optimality? (Asymptotic)

$$\mathbb{E}[\|\hat{\theta}(X) - \theta\|^2] \geq \mathrm{CRLB} \qquad \frac{\mathbb{E}[\|\hat{\theta}_{\mathrm{MLE}}(X) - \theta\|^2]}{\mathrm{CRLB}} \to 1$$

(CRAMER-RAO LOWER BOUND)

$|X| \to \infty$

Data - Driven Version

In the classical setup, parameter estimation makes sense only if there is parametric density function $\mathsf{f}(X|\theta)$

We will sacrifice generality in order to define a meaningful parameter estimation problem that can be formulated under a data-driven setup

We will define $\mathsf{f}(X|\theta)$ indirectly!!

We start with random vector $\mathcal{Z} \sim \mathsf{g}(Z)$

We consider a deterministic transformation $\mathsf{T}(Z,\theta)$ which is known up the some parameter vector θ

We define the random vector $\mathcal{X} = \mathsf{T}(\mathcal{Z},\theta)$ which has density $\mathcal{X} \sim \mathsf{f}(X|\theta)$

Assume instead of $\mathsf{g}(Z)$ we have dataset $\{Z_1, \ldots, Z_m\}$ with independent realizations of $\mathcal{Z}$

Assume that we are given a dataset $\{X_1, \ldots, X_n\}$ with independent realizations of $\mathcal{X}$ all following $\mathsf{f}(X|\theta)$ with the same θ.

Problem:

Using $\{Z_1, \ldots, Z_m\}$ as a representative of the density $\mathsf{g}(Z)$

Assuming knowledge of the transformation $\mathsf{T}(Z, \theta)$ up to the unknown parameters θ

For every collection of data $\{X_1, \ldots, X_n\}$ estimate the θ that has generated them.

IF for the data we had $X_i = \mathsf{T}(Z_i, \theta)$ the problem would have been simple.

We could form some distance between the Xs and the $\mathsf{T}(Z, \theta)$s and minimize over θ

BUT the two datasets $\{Z_1, \ldots, Z_m\}$ and $\{X_1, \ldots, X_n\}$ are independent and unrelated.

Moment Matching

If $\mathcal{X} = \mathsf{T}(\mathcal{Z}, \theta)$ then

$$\mathbb{E}[\mathcal{X}^{\cdot p}] = \mathbb{E}\left[\left(\mathsf{T}(\mathcal{Z}, \theta)\right)^{\cdot p}\right]$$

$$\boxed{\frac{1}{n}\sum_{i=1}^{n}(X_i)^{\cdot p} = \frac{1}{m}\sum_{j=1}^{m}\left(\mathsf{T}(Z_j, \theta)\right)^{\cdot p}}$$

$p = p_1, p_2, \ldots$

Sufficient # of equations to solve for θ.

Many different choices for moments

Moment estimates are Notoriously NON-ROBUST

Density Matching

We would like to find θ so that $\mathcal{X}$ and $\mathsf{T}(\mathcal{Z},\theta)$ exhibit the same statistical behavior

We would like to find θ so that $\{X_1,\ldots,X_n\}$ and $\{\mathsf{T}(Z_1,\theta),\ldots,\mathsf{T}(Z_m,\theta)\}$ exhibit the same statistical behavior

There exists an interesting methodology developed for Generative Modeling and our problem constitutes a special case.

Generative Adversarial Networks (GANs)

The random vector $\mathcal{X}$ follows $\mathsf{f}(X)$ and the random vector $\mathcal{Z}$ follows $\mathsf{g}(Z)$

We would like to find a transformation (generator) $\mathsf{G}(Z)$ such that $\mathcal{Y} = \mathsf{G}(\mathcal{Z})$ follows $\mathsf{f}(X)$

To solve the problem we are going to design a second function $\mathsf{D}(X)$ (discriminator) by considering the adversarial problem

$$\min_{\mathsf{G}(Z)} \max_{\mathsf{D}(X)} J(\mathsf{D}, \mathsf{G})$$

2014 Goodfellow et al.

where

$$J(\mathsf{D}, \mathsf{G}) = \mathbb{E}_{\mathcal{X}}\left[\log \mathsf{D}(\mathcal{X})\right] + \mathbb{E}_{\mathcal{Z}}\left[\log\left(1 - \mathsf{D}(\mathsf{G}(\mathcal{Z}))\right)\right]$$

$\rightarrow G(z)$ such that $\mathcal{Y} = G(\mathcal{Z}) \sim f(x)$

(17)

Extensions

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

$\psi'(z) < 0$
$\omega'(r) > 0$
$\phi'(z) = -\omega^{-1}(z)\psi'(z)$

$$\min_{\mathsf{G}(Z)} \max_{\mathsf{D}(X)} J(\mathsf{D},\mathsf{G})$$

designs the correct "generator"

Data-Driven Setup

Under a data-driven setup $\mathsf{G}(Z) \to \mathsf{G}(Z,\theta)$ and $\mathsf{D}(X) \to \mathsf{D}(X,\vartheta)$ with the generator and discriminator becoming parametric transformations

This is exactly the same with our problem where the "generator" is $\mathsf{T}(Z,\theta)$.

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

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

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

If Z realization of $\mathcal{Z}$ following $\mathsf{g}(Z)$ then $\mathsf{G}(Z,\theta_o)$ realization of $\mathcal{X}$ following $\mathsf{f}(X)$.

EXAMPLE

If $\{X_1, X_2, \ldots, X_n\}$ (many) human faces and
$\{Z_1, Z_2, \ldots, Z_m\}$ (many) Gaussian random vectors

Can design function $G(Z,\theta_o)$ such that when Z Gaussian vector $G(Z,\theta_o)$ is A HUMAN FACE

DATASET CELIBA $(x_1, x_2, \dots)$

True faces

Generate $z_1, z_2, \dots$ Gaussian vectors

$G(z_1, \theta_0)$ $G(z_2, \theta_0)$ - - - -

Synthetic faces

Assume a random vector $\mathcal{X}$ is described by a Generative Model: $\mathsf{g}(Z), \mathsf{G}(Z)$ instead of a density $\mathsf{f}(X)$

Advantage: If we can easily generate realizations of $\mathsf{g}(Z)$ then we transform them and generate realizations of $\mathsf{f}(X)$

Inverse Problems

For $\mathcal{X}$ we have generative model $\mathsf{g}(Z), \mathsf{G}(Z)$

If X, a realization of $\mathcal{X}$, undergoes a transformation $Y = \mathsf{F}(X)$ then X can be recovered from Y by first recovering the Z that generates X as $X = \mathsf{G}(Z)$.

X

Y

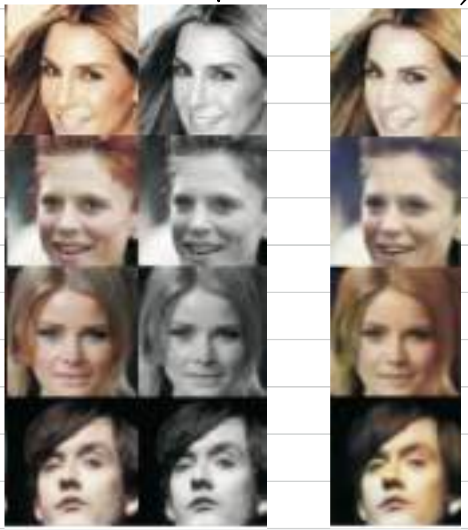
$X = G(Z) \qquad Z \sim g(z)$

Gaussian.

from Y estimate $\hat{Z}$

then $\underline{\hat{X} = G(\hat{Z})}$

X Y $\hat{X} = G(\hat{z})$



Colorization

X

Y

$\hat{X} = G(\hat{z})$

De-Quantization

X^1 X^2 Y $\hat{X}^1 = G_1(\hat{z}^1)$ $\hat{X}^2 = G_2(\hat{z}^2)$

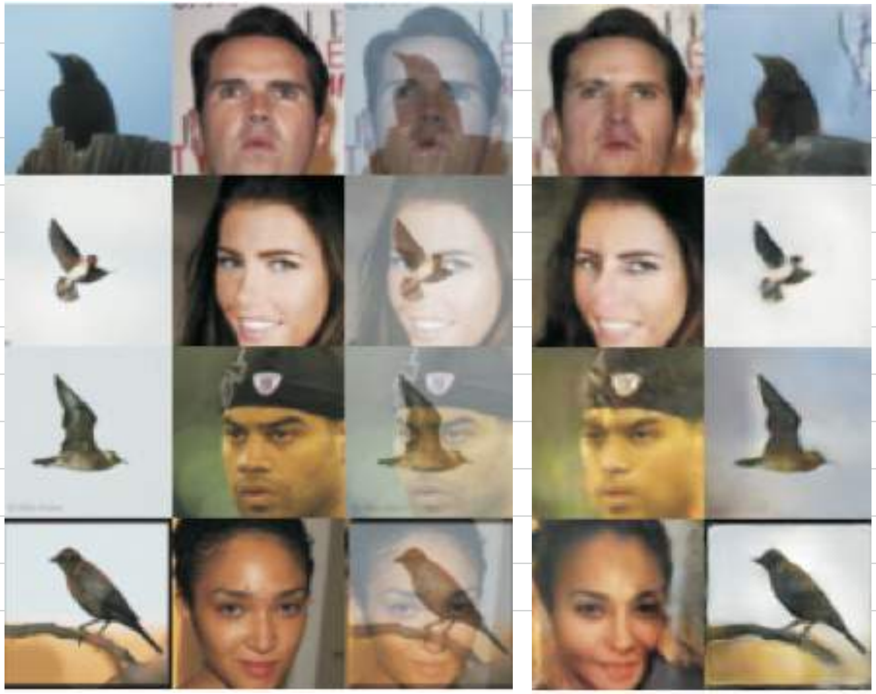


Image Seperation