

Deep Generative Models in Vision: An Introduction

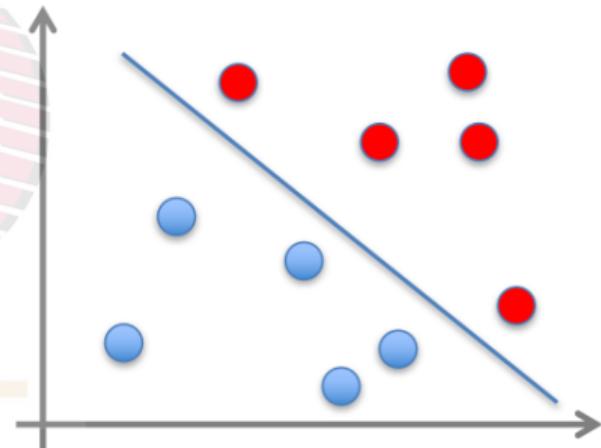
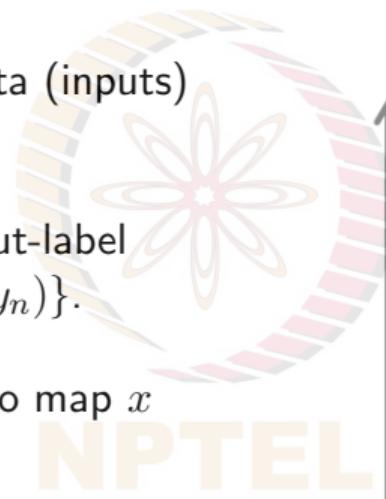
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad



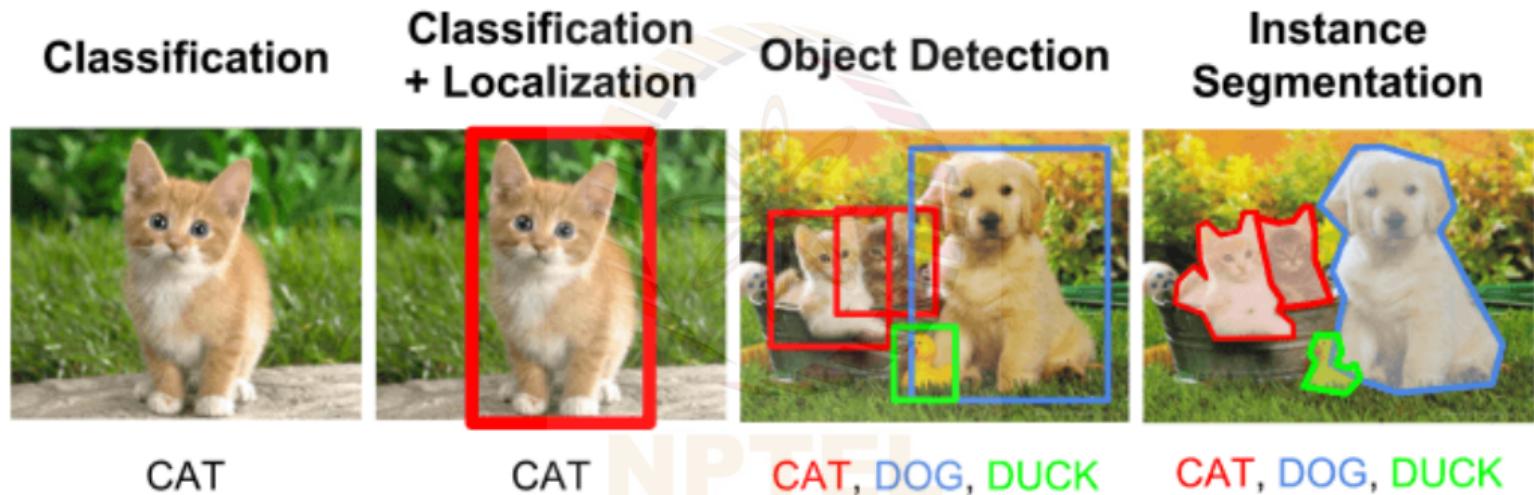
Supervised Learning

- Learning a mapping between data (inputs) to label (output).
- Data is given in the form of input-label pairs $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.
- The goal is to learn a function to map x to y i.e., $f(x) = y$.



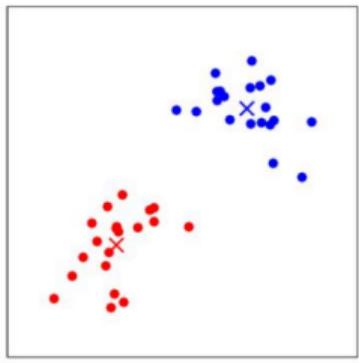
Let's see some examples!

Supervised Learning: Examples in Computer Vision

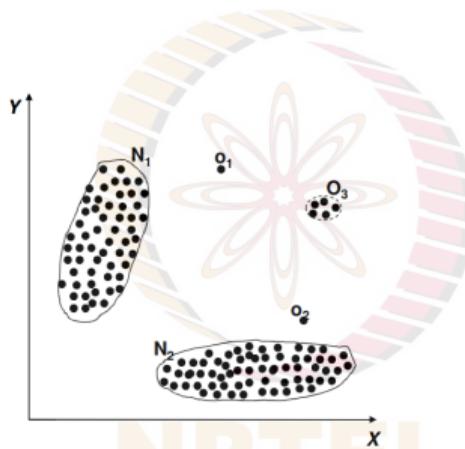


Credit: Fei Fei Li, J Johnson and S Yeung, CS231n, Stanford Univ

Going Beyond Supervised Learning



Unsupervised Learning/ Data Understanding



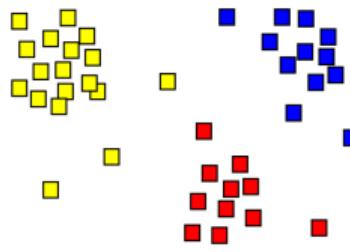
Detecting Outliers

7	3	9	3	9	9
1	1	0	6	0	0
0	1	9	1	2	2
6	3	2	0	8	8

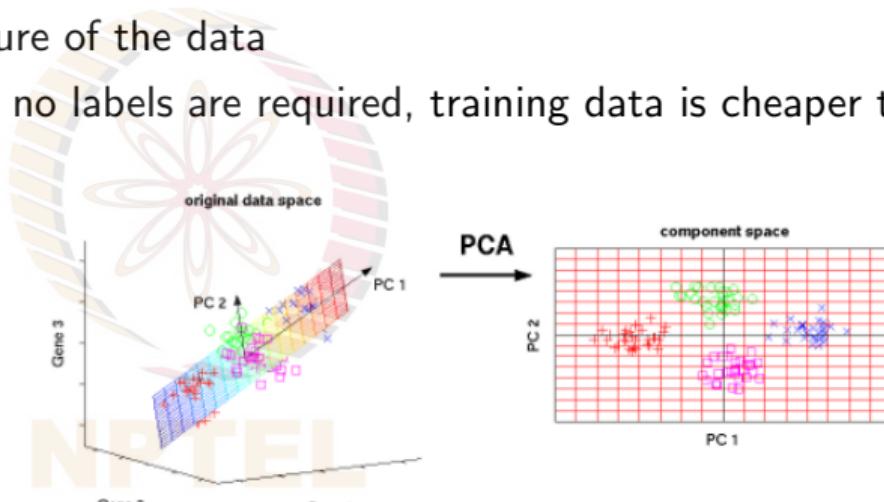
Generating Data

Unsupervised Learning

- Capturing the underlying structure of the data
- Only data (x) is provided; since no labels are required, training data is cheaper to obtain



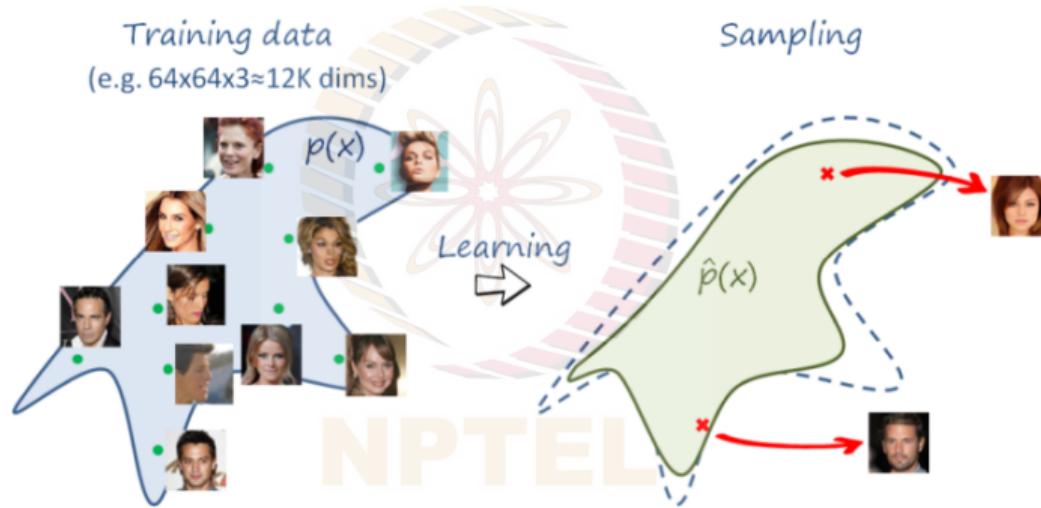
Clustering



Dimensionality Reduction

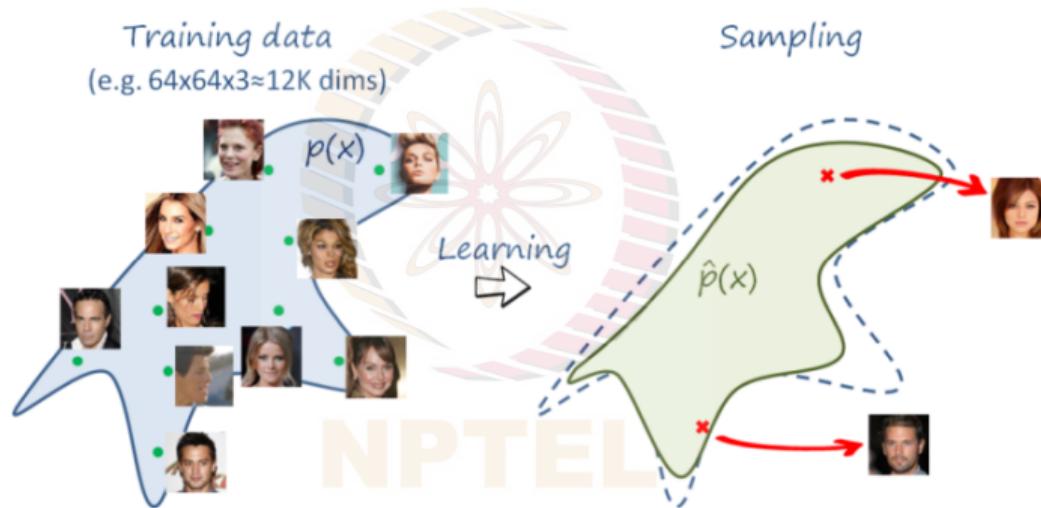
Generative Models

- **Goal:** Generate data samples similar to the ones in the training set



Generative Models

- **Goal:** Generate data samples similar to the ones in the training set



- Assume training data $X = \{x_1, x_2, \dots, x_n\}$ comes from an underlying distribution $p_D(x)$, and a generative model samples data from a distribution $p_M(x)$
- Our aim is to **minimize** some notion of **distance** between $p_D(x)$ and $p_M(x)$

Generative Models: How to learn?

Aim: To minimize some notion of distance between $p_D(x)$ and $p_M(x)$; how?

- Given a dataset $X = x_1, x_2, x_3, \dots, x_N$ from an underlying distribution $p_D(x)$



Generative Models: How to learn?

Aim: To minimize some notion of distance between $p_D(x)$ and $p_M(x)$; how?

- Given a dataset $X = x_1, x_2, x_3, \dots, x_N$ from an underlying distribution $p_D(x)$
- Consider an approximating distribution $p_M(x)$ coming from a family of distributions M , i.e. we need to find the best distribution in M , parametrized by θ , which minimizes distance between p_M and p_D , i.e.:

$$\theta^* = \arg \min_{\theta \in M} \text{dist}(p_\theta, p_D)$$

NPTEL

Generative Models: How to learn?

Aim: To minimize some notion of distance between $p_D(x)$ and $p_M(x)$; how?

- Given a dataset $X = x_1, x_2, x_3, \dots, x_N$ from an underlying distribution $p_D(x)$
- Consider an approximating distribution $p_M(x)$ coming from a family of distributions M , i.e. we need to find the best distribution in M , parametrized by θ , which minimizes distance between p_M and p_D , i.e.:

$$\theta^* = \arg \min_{\theta \in M} \text{dist}(p_\theta, p_D)$$

- What distance to choose?

NPTEL

Generative Models: How to learn?

Aim: To minimize some notion of distance between $p_D(x)$ and $p_M(x)$; how?

- Given a dataset $X = x_1, x_2, x_3, \dots, x_N$ from an underlying distribution $p_D(x)$
- Consider an approximating distribution $p_M(x)$ coming from a family of distributions M , i.e. we need to find the best distribution in M , parametrized by θ , which minimizes distance between p_M and p_D , i.e.:

$$\theta^* = \arg \min_{\theta \in M} \text{dist}(p_\theta, p_D)$$

- What distance to choose?
- If KL-divergence is the distance function, the above problem becomes one of maximum likelihood estimation!

$$\theta^* = \arg \min_{\theta \in M} \mathbb{E}_{x \sim p_D} [-\log p_\theta(x)]$$

Generative Models: How to learn?

Aim: To minimize some notion of distance between $p_D(x)$ and $p_M(x)$; how?

- Given a dataset $X = x_1, x_2, x_3, \dots, x_N$ from an underlying distribution $p_D(x)$
- Consider an approximating distribution $p_M(x)$ coming from a family of distributions M , i.e. we need to find the best distribution in M , parametrized by θ , which minimizes distance between p_M and p_D , i.e.:

$$\theta^* = \arg \min_{\theta \in M} \text{dist}(p_\theta, p_D)$$

- What distance to choose?
- If KL-divergence is the distance function, the above problem becomes one of maximum likelihood estimation!

$$\theta^* = \arg \min_{\theta \in M} \mathbb{E}_{x \sim p_D} [-\log p_\theta(x)]$$

Why? **Homework!**

- This is the idea in a few methods (e.g. PixelCNN/PixelRNN, which we will see later)

Generative Models: Applications



bicubic
(21.59dB/0.6423)



SRResNet
(23.53dB/0.7832)



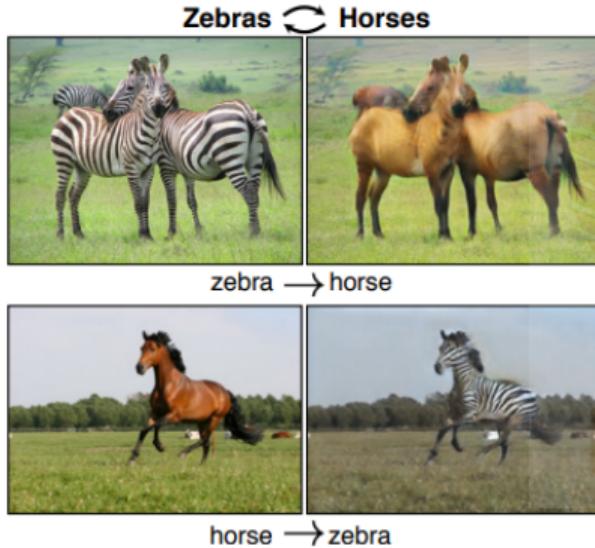
SRGAN
(21.15dB/0.6868)



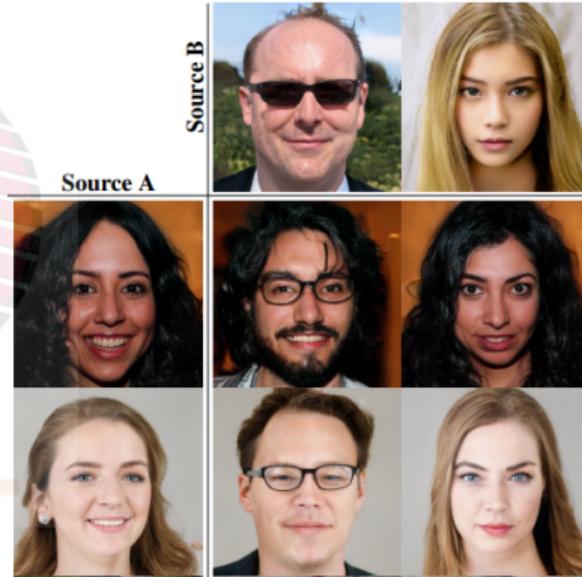
Image Super Resolution

Image Colorization

Generative Models: Applications



Cross-domain Image Translation



Generating Realistic Face Datasets

Generative Models: More Applications

- Learn good generalizable latent features



Generative Models: More Applications

- Learn good generalizable latent features
- Augment small datasets



Generative Models: More Applications

- Learn good generalizable latent features
- Augment small datasets
- Enable mixed-reality applications such as Virtual Try-on



NPTEL

Generative Models: More Applications

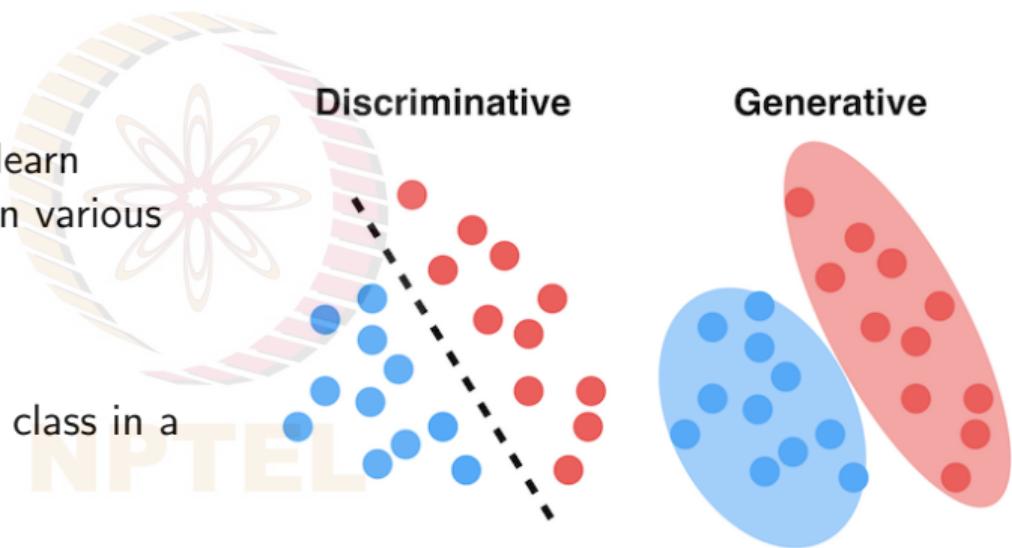
- Learn good generalizable latent features
- Augment small datasets
- Enable mixed-reality applications such as Virtual Try-on
- Many more...



NPTEL

What are Generative Models

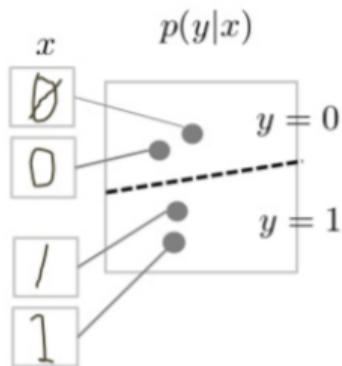
- **Discriminative** models: aim to learn **differentiating features** between various classes in a dataset
- **Generative** models aim to learn **underlying distribution** of each class in a dataset



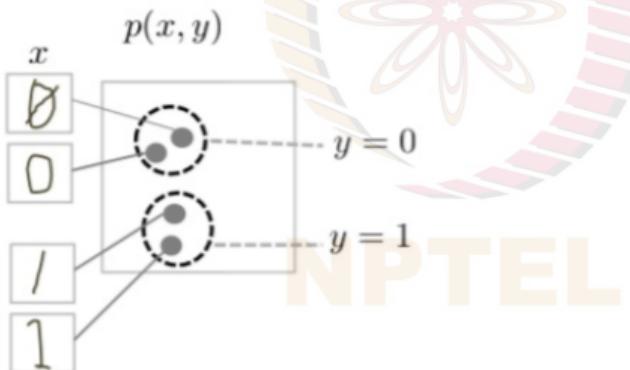
Discriminative vs Generative Models

- Consider a binary classification problem of classifying images of 1s and 0s

- Discriminative Model

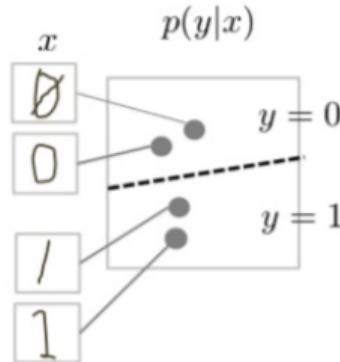


- Generative Model

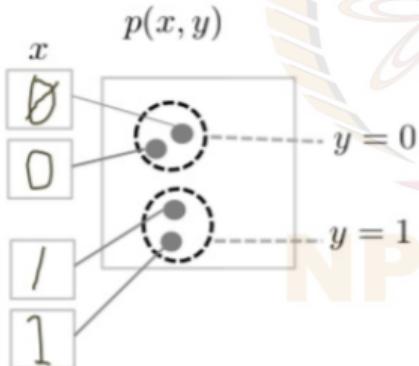


Discriminative vs Generative Models

- Discriminative Model



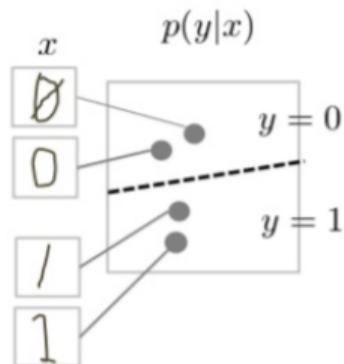
- Generative Model



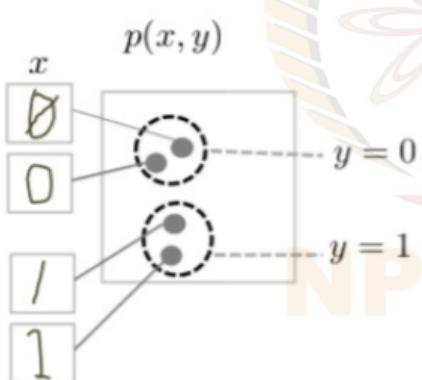
- Consider a binary classification problem of classifying images of 1s and 0s
- A **discriminative classifier** directly models the **posterior** i.e. $p(y|x)$; x is always given as input

Discriminative vs Generative Models

- Discriminative Model



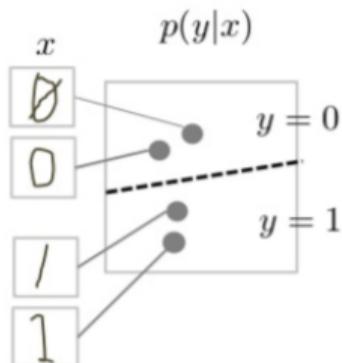
- Generative Model



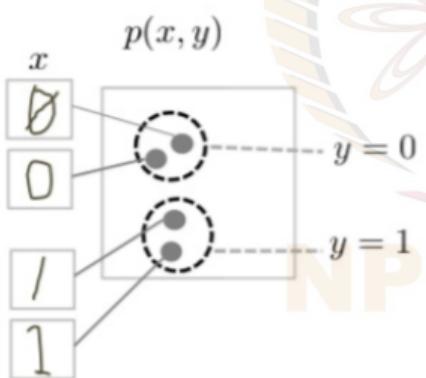
- Consider a binary classification problem of classifying images of 1s and 0s
- A **discriminative classifier** directly models the **posterior** i.e. $p(y|x)$; x is always given as input
- A **generative classifier** models the **joint distribution** i.e. $p(x,y)$

Discriminative vs Generative Models

- Discriminative Model



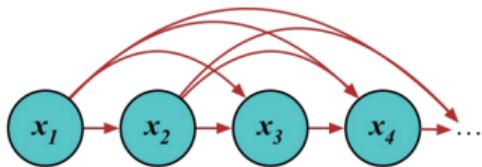
- Generative Model



- Consider a binary classification problem of classifying images of 1s and 0s
- A **discriminative classifier** directly models the **posterior** i.e. $p(y|x)$; x is always given as input
- A **generative classifier** models the **joint distribution** i.e. $p(x,y)$
- Recall: posterior and joint are related as:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$$

Generative Models

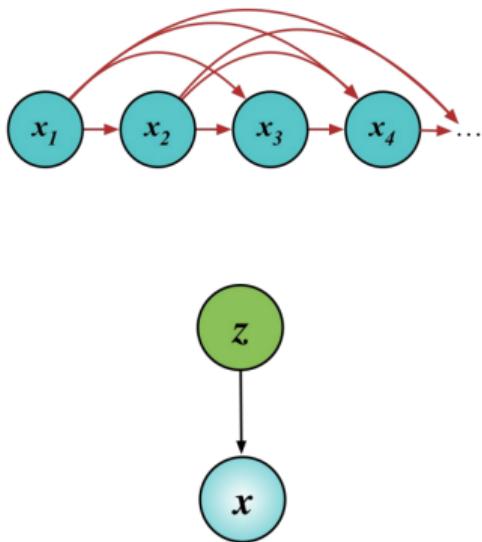


Two main kinds of generative models:

- ① **Fully Visible Models:** Directly model observations without introducing extra variables; e.g. considering each pixel value of an image as an observation

NPTEL

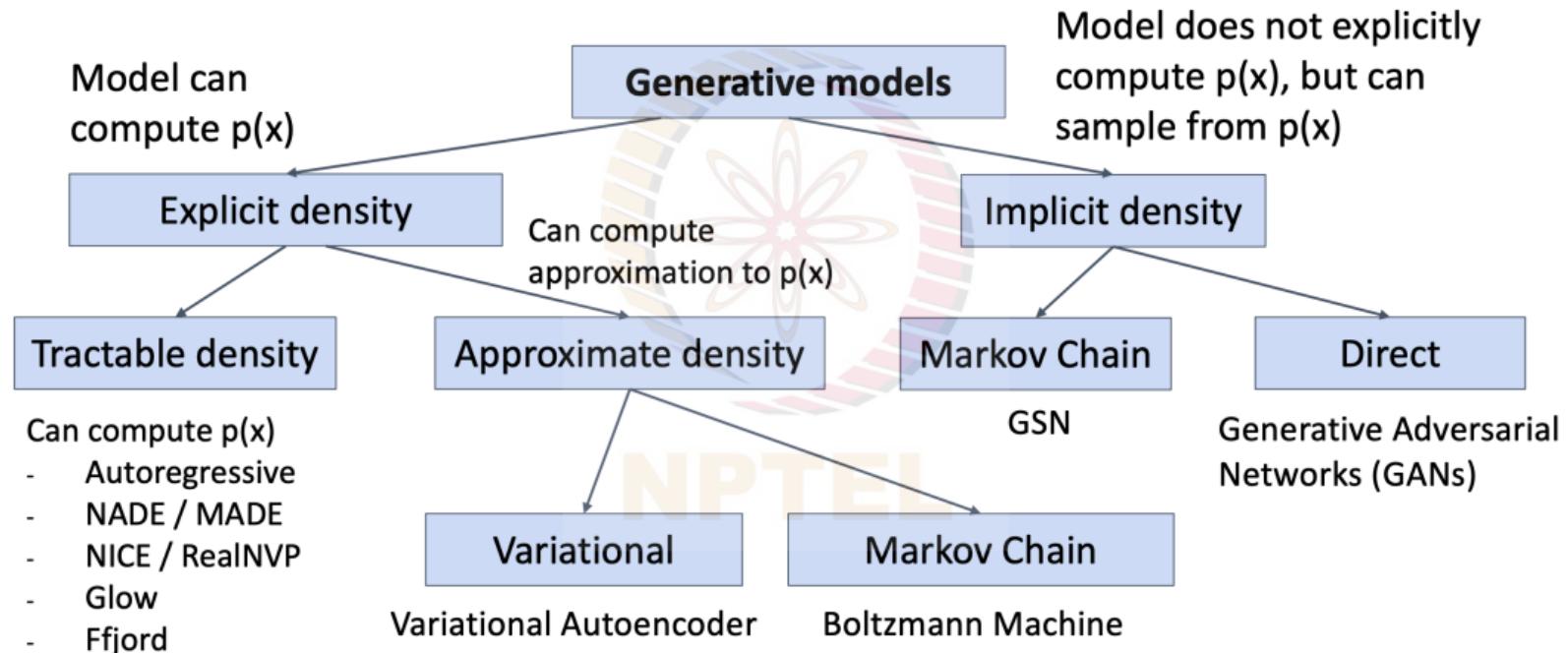
Generative Models



Two main kinds of generative models:

- ① **Fully Visible Models:** Directly model observations without introducing extra variables; e.g. considering each pixel value of an image as an observation
- ② **Latent Variable Models:** Defining hidden variables which generate observed data:
 - ① **Explicit Likelihood Estimation Models:** Explicitly define and learn likelihood of data; e.g. Variational Autoencoders (VAEs)
 - ② **Implicit Models:** Learn to directly generate samples from model's distribution, without explicitly defining any density function; e.g. Generative Adversarial Networks (GANs)

Taxonomy of Generative Models



Credit: Fei-Fei Li, J Johnson, CS231n, Stanford Univ

Homework

Readings

- Aditya Grover, Stefano Ermon, Tutorial on Deep Generative Models, IJCAI-ECAI 2018
- (Optional) Shakir Mohamed, Danilo Rezende, Deep Generative Models Tutorial, UAI 2017

Exercise

- Why does using KL-divergence in finding the generative model simplify to maximum likelihood estimation?