Generative Models

Dr. Konda Reddy Mopuri
Dept. of AI
IIT Hyderabad

Typical (broad) categorization of ML tasks

Supervised

- Driving data: samples (x,y)
- x is data, y is its label
- Aim: mapping function $x \rightarrow y$
- E.g. Classification, Regression, Object Detection, Semantic Segmentation, etc.

Unsupervised

- Driving data: samples x
- No labels
- Learn 'some structure' from the data
- E.g. Clustering, Dimensionality Reduction, Feature Learning, Density Estimation, etc.

Another categorization

- Discriminative vs. Generative Models

Generative vs. Discriminative

Discriminative

- Learn the boundaries between the classes
- Learns $p(y/x)$

Generative

- Model the distribution of individual classes
- Learns $p(x)$
- Conditional Generative models learn $p(x/y)$

Probability density function (PDF)

- Function on the sample space that indicates relative likelihood of the random variable
- Non-negative function
- Normalized to 1 (AUC)
- → Different values of random variable (x) compete for the density
- Discrete random variable → PMF

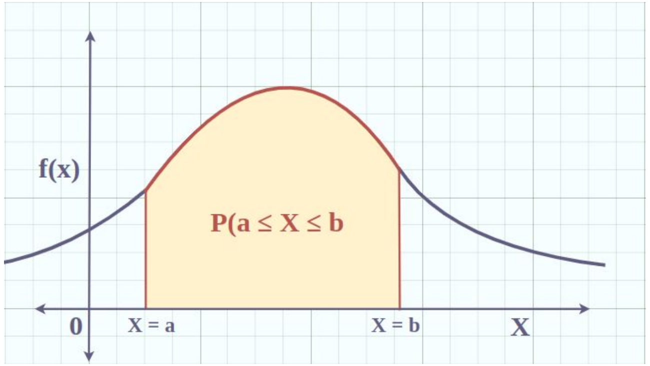


Figure Credits: GeeksforGeeks

Generative model

- Given training data, generate samples from the same distribution

learning → $p_{model}(x)$ → sampling

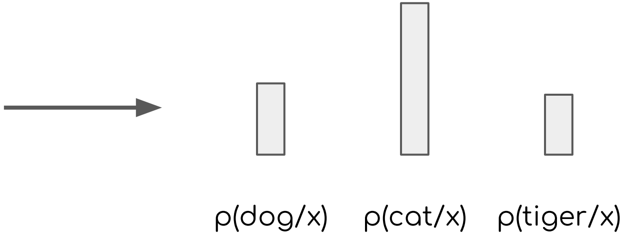
Training data ~ p_{data} (x)

Generative vs. Discriminative

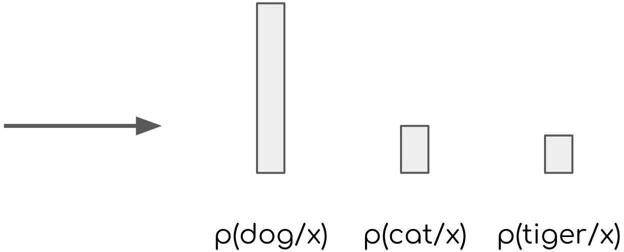
Discriminative

- Learns p(y/x)
- Competition among the set of labels for each input (not across inputs)

x



x

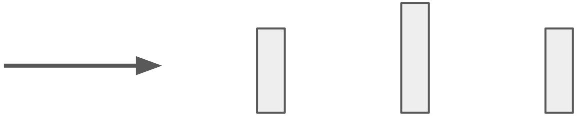


Generative vs. Discriminative

Discriminative

- Must predict labels for any input

x

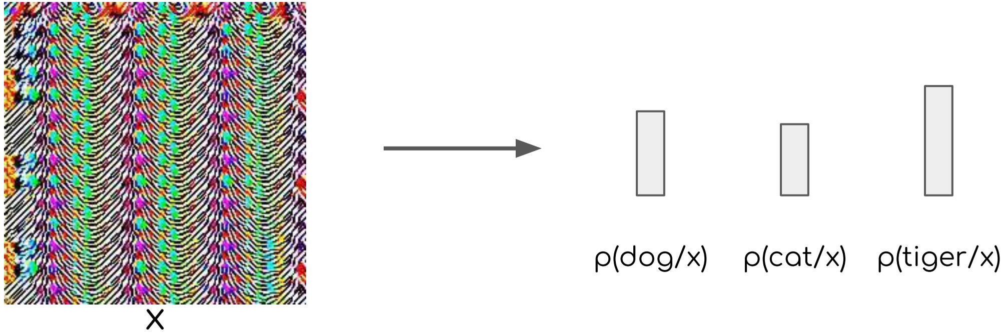


p(dog/x) p(cat/x) p(tiger/x)

Generative vs. Discriminative

Discriminative

- Can't reject inputs!

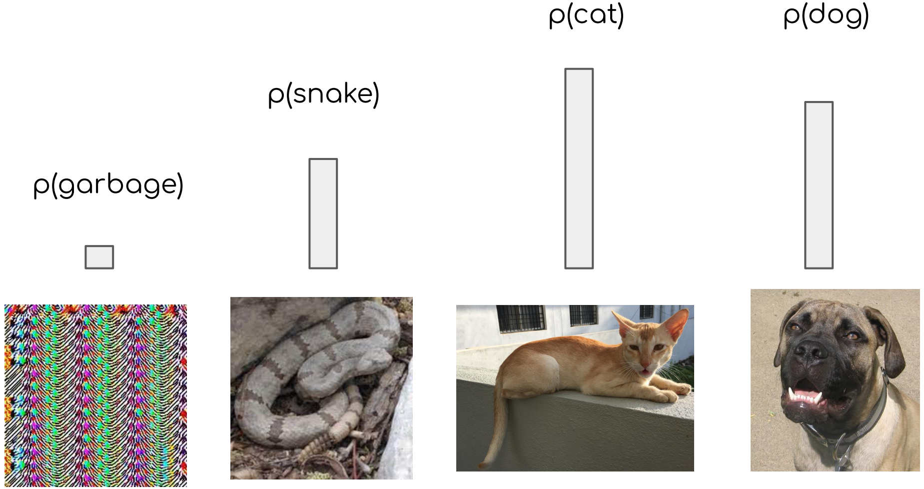


Generative vs. Discriminative

Generative

- Learns p(x)
- Competition is among different samples

Requires a great understanding of the data (e.g., images)
How likely for a snake to be on the ground? Or, in the air? Or, next to a human?

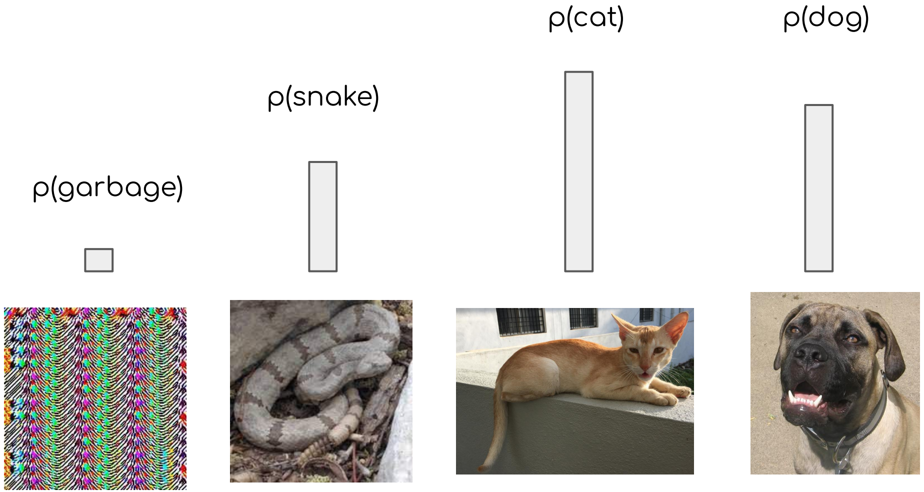


Generative vs. Discriminative

Generative

- Learns p(x)
- Competition is among different samples

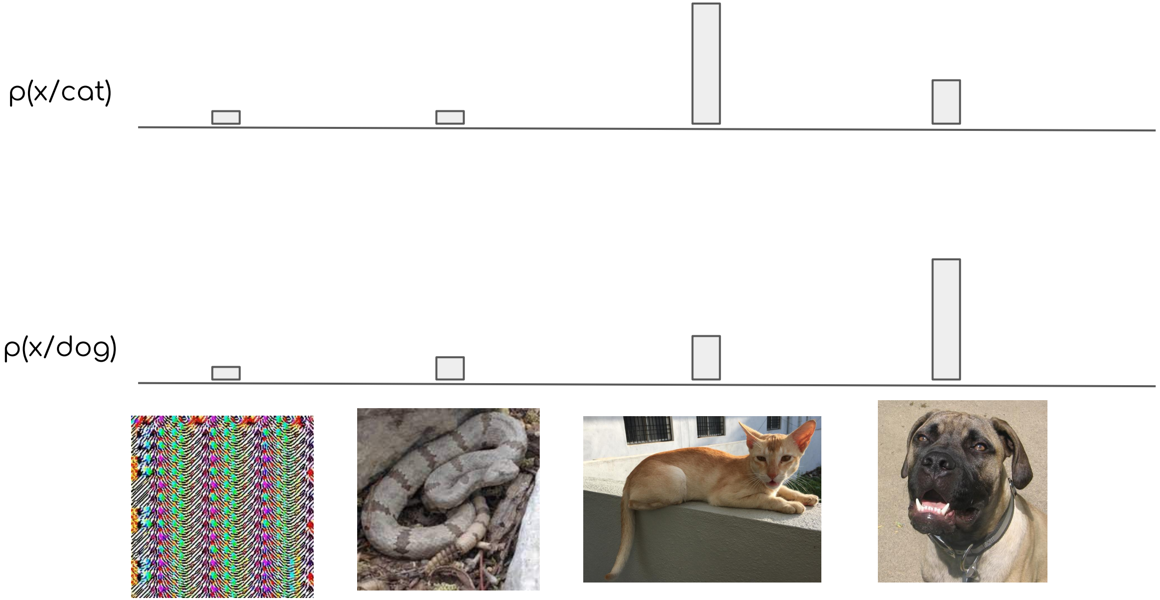
Importantly, the model can reject unreasonable samples as 'unlikely' (small value assignment)



Generative vs. Discriminative

Conditional Generative

- Learns p(x/y)
- Conditioning label results in competition among different samples



Generative vs. Discriminative

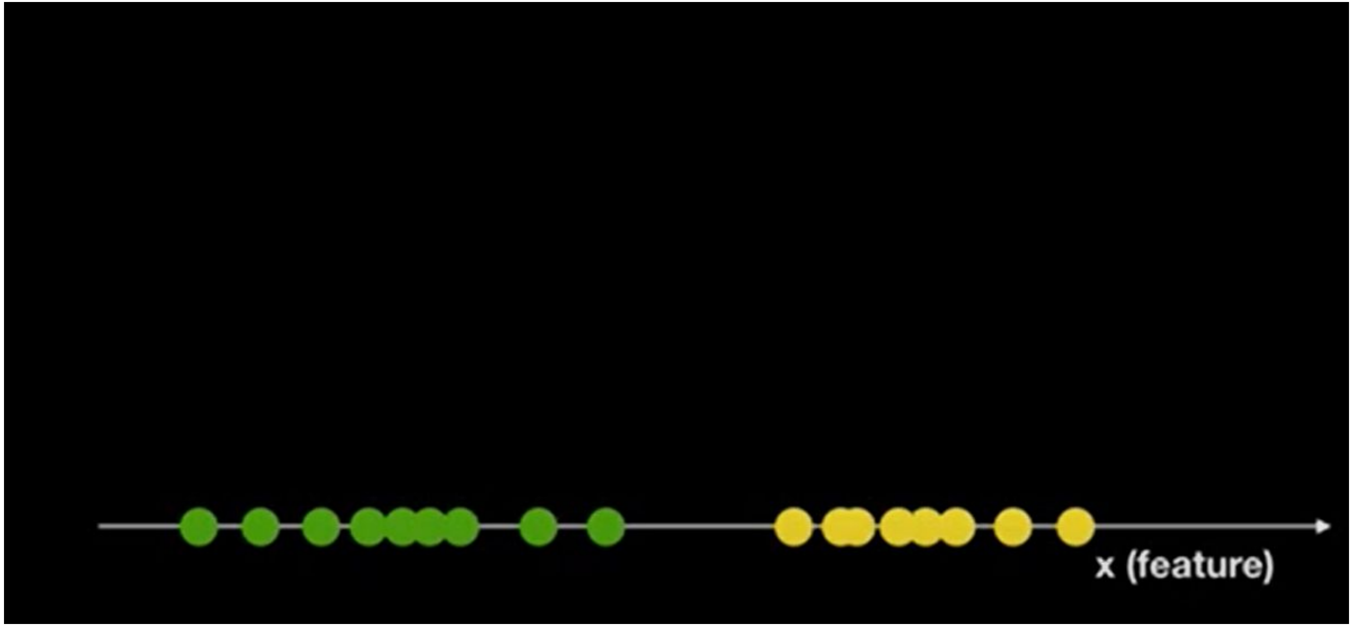
- Conditional generative models can be built from other components!

$$\overbrace{\boxed{p(x/y)}}^{\text{Conditional generative model}} = \frac{\overbrace{\boxed{p(y/x)}}^{\text{Discriminative model}}}{\underbrace{\boxed{p(y)}}_{\text{Label prior}}} \underbrace{\boxed{p(x)}}_{\text{Unconditional generative model}}$$

Summary

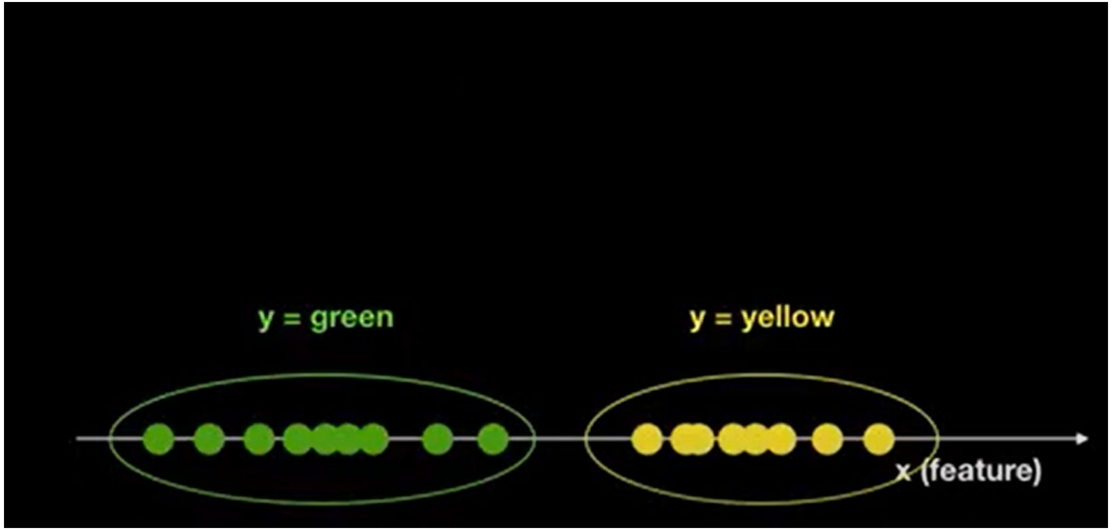
- Discriminative $p(y/x)$ → assign labels, feature learning (with labels)
- Generative $p(x)$ → Detect outliers, feature learning (w/o labels), and sample to generate new data!
- Conditional $p(x/y)$ → Assign labels and detect outliers, Generate new data conditioned on the label!

Example



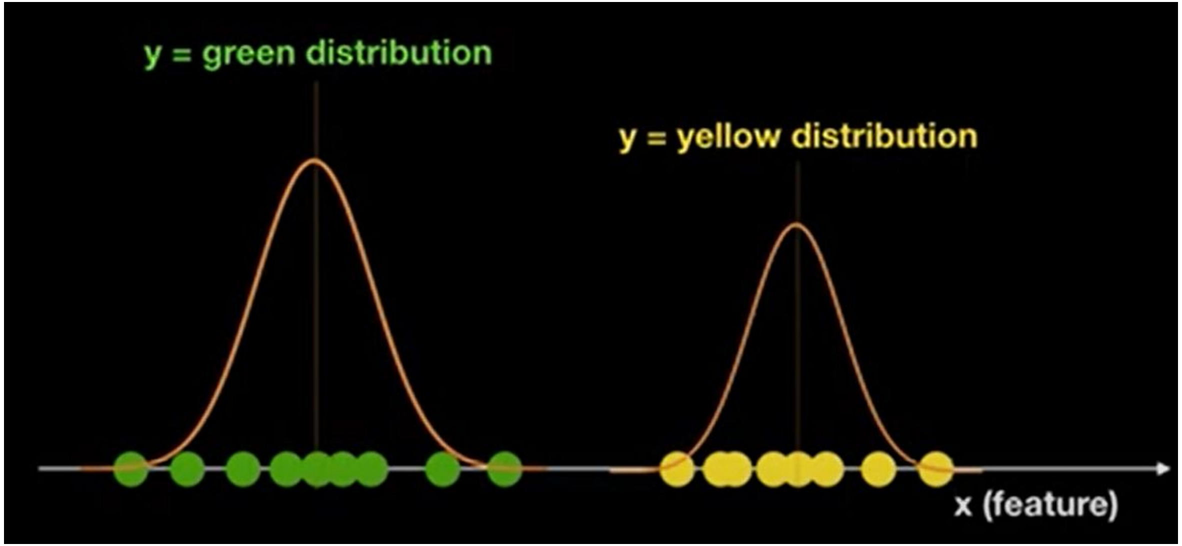
Conditional Generative Model

- Given data, model estimates distribution of p(x/y)



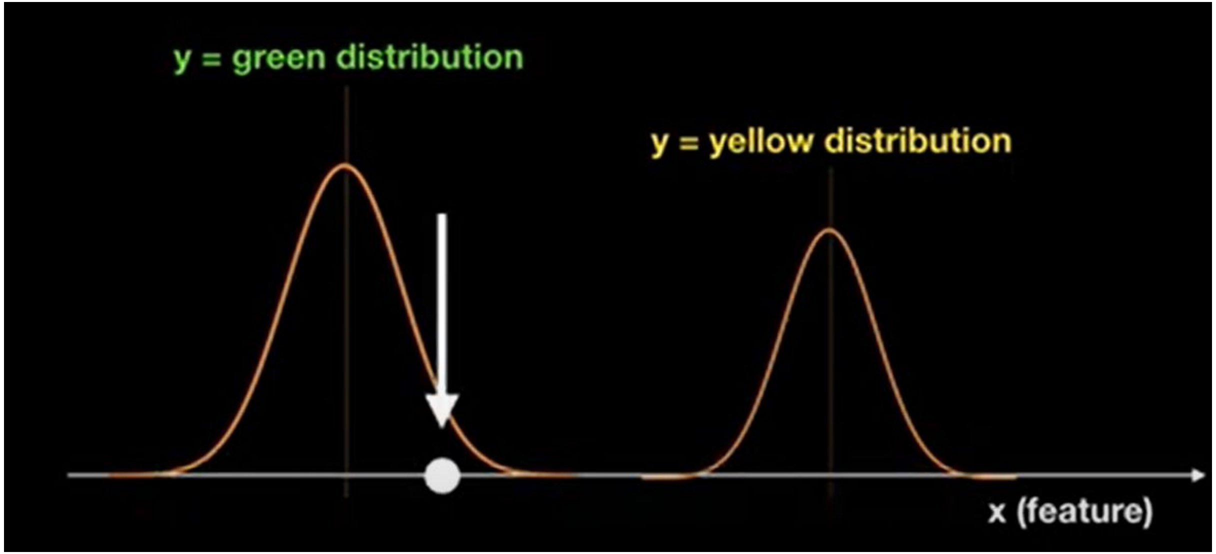
Conditional Generative Model

- Given data, model estimates distribution of p(x/y)



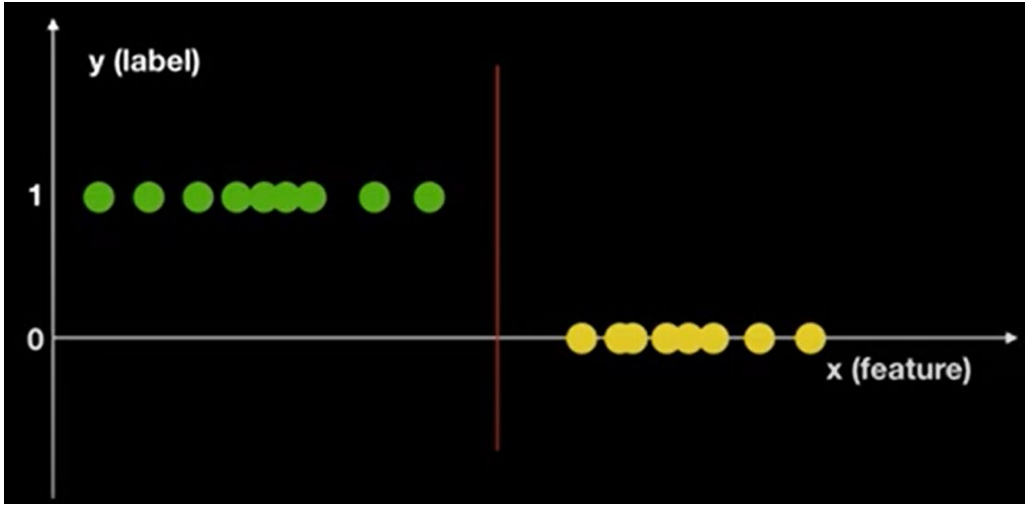
Conditional Generative Model

- What is the value of y for new data?



Discriminative Model

- Focus is on 'How to distinguish different classes?'
- Goal is to find the decision boundary



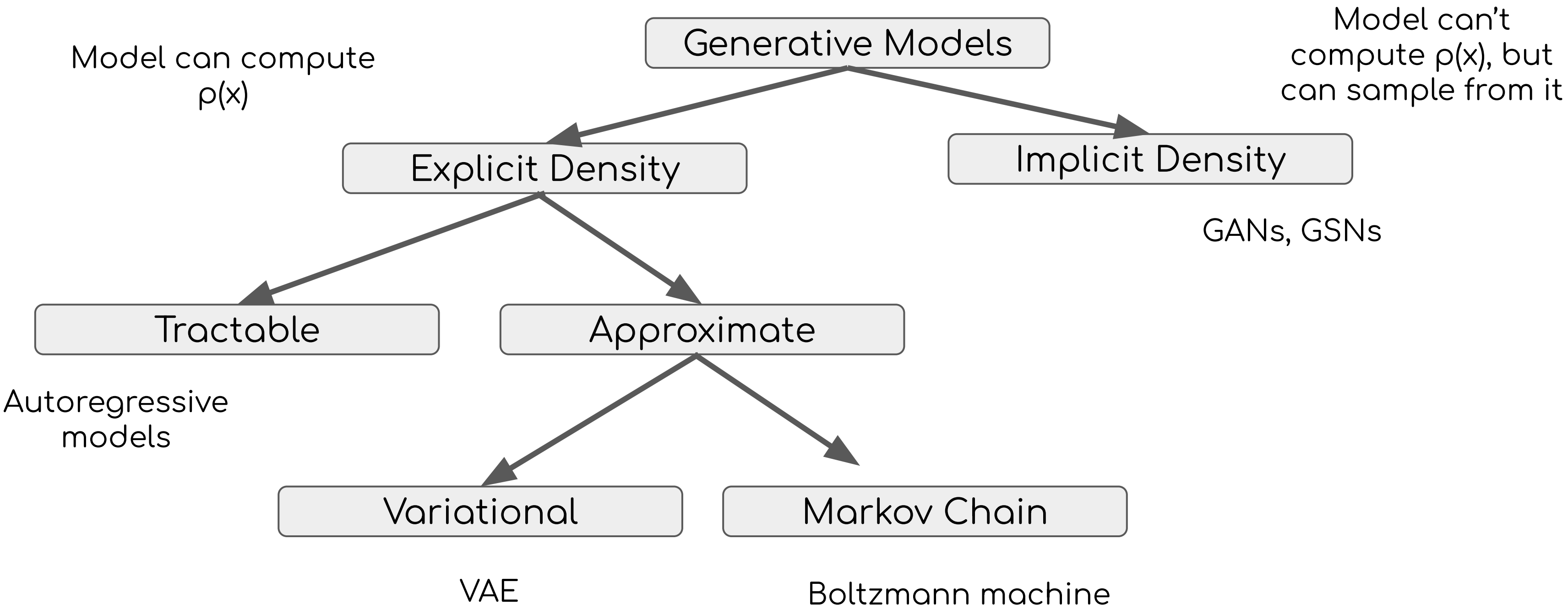
Discriminative Model

- Focus is on 'How to distinguish different classes?'
- Goal is to find the decision boundary
- Uses $p(y/x)$ to classify → predicts the class with highest $p(y/x)$

Why generative models?

- Real-like samples for artwork
- Useful transformations - colorization, inpainting, super-resolution
- Representation learning (e.g., for discriminative tasks)
- Provides insights about the high-dim data
- Modeling physical world for simulation & planning (e.g., Robotics and RL)

Broad categorization of Generative Models



Models with Explicit Density

$$p(x) = f(x, \theta)$$

- Given the training data $\{x_1, x_2, \ldots. x_N\}$, train the model via maximum likelihood estimation (MLE)

$$\theta^* = \arg\max_\theta \prod_i p(x_i)$$

$$\theta^* = \arg\max_\theta \sum_i log \;\; p(x_i)$$

$$\theta^* = \arg\max_\theta \sum_i log \;\; f(x_i, \theta)$$

Autoregressive models

- Based on the assumption that sample x consists of multiple portions (subparts) → $x = (x_1, x_2, x_3, \ldots.. x_W)$
- Probability is expressed using the chain rule

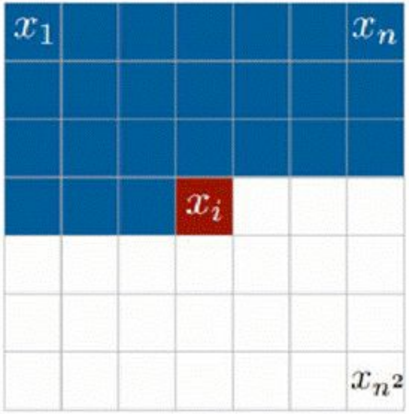
$$p(x_1, x_2, x_3, \ldots.. x_W) = p(x_1)\, p(x_2/x_1) p(x_3/x_2,x_1) \ldots\ldots = \Pi\, p(x_t/x_1 x_2 \ldots x_{t-1})$$

→ Recurrent models (RNNs) can do this!

Autoregressive models: Pixel RNNs (ICML 2016)

- Generate the pixels of an image: one at a time, from top-left to bottom right
- Hidden state of each pixel is modeled from the hidden state and pixel values of left and top pixels

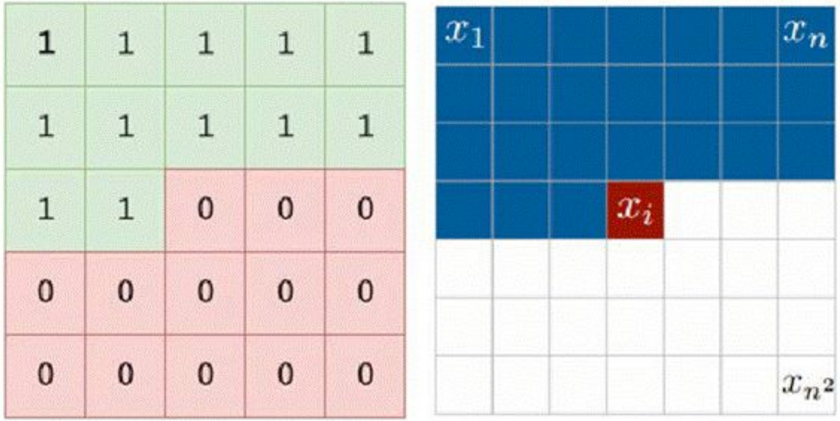
1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0



$$h_{i,j} = f(h_{i-1,j}, h_{i,j-1}, \theta)$$

Autoregressive models: Pixel RNNs (ICML 2016)

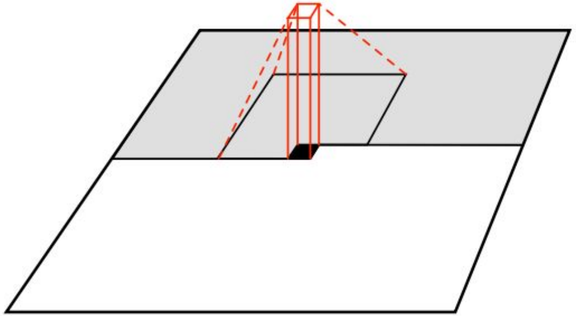
- Very slow during both training and testing!



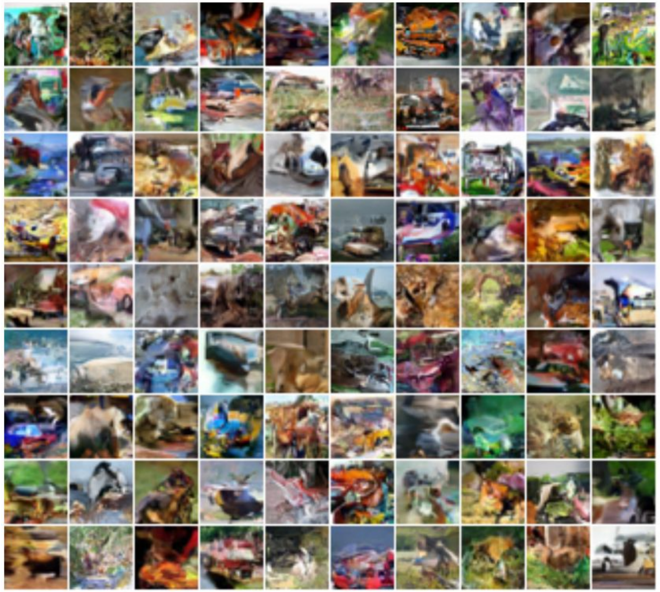
$$h_{i,j} = f(h_{i-1,j}, h_{i,j-1}, \theta)$$

Autoregressive models: Pixel CNNs (NeurIPS 2016)

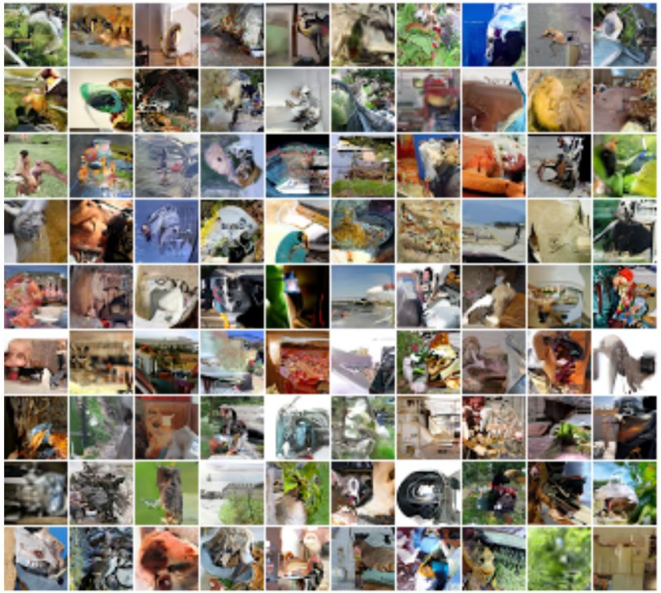
- Dependency is modeled with a CNN over the context
- Training MLE
- Training is faster but sampling is slow



Pixel RNN results

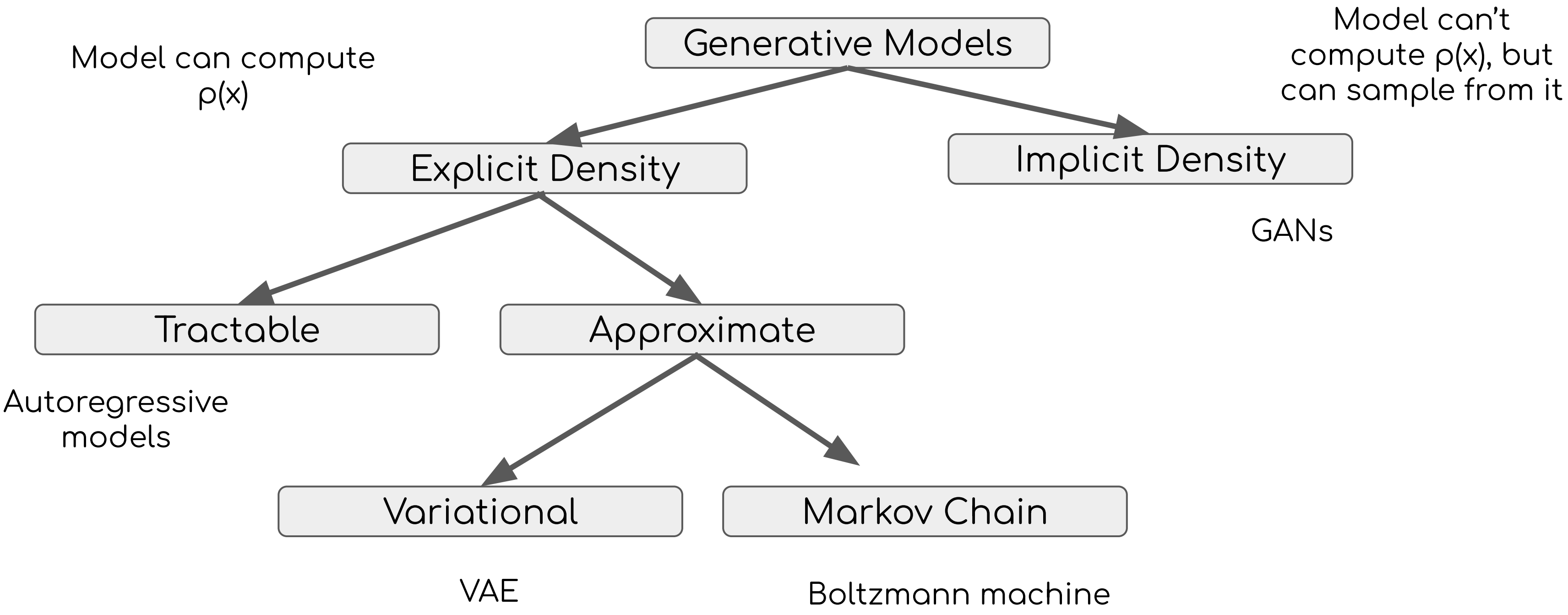


CIFAR-10 (32 X 32)



ImageNet (32 X 32)

Broad categorization of Generative Models



Variational Autoencoders (VAE)

VAE

- Deal with an intractable density → can't be computed/optimized explicitly
- Optimizes the 'lower bound' on the density

Next: VAEs