

Lecture 1: Introduction

Deep Generative Models

Sajjad Amini

Department of Electrical Engineering
Sharif University of Technology

Contents

- 1 Supervised vs. Unsupervised Learning
- 2 Unconditional Generative Models
- 3 Conditional Generative Models
- 4 Discriminative Vs. Generative Classification
- 5 Big Picture
- 6 Conclusions

Section 1

Supervised vs. Unsupervised Learning

Supervised Learning

Supervised Learning

- **Observations:** $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$

- **Applications:**

- Classification
- Regression
- ...

- **Model:** $p_{\theta}(y|\mathbf{x})$

- **Phases:**

- *Training Phase:*

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{n=1}^N \text{Loss}\left(y_n, p_{\theta}(y|\mathbf{x}_n)\right)$$

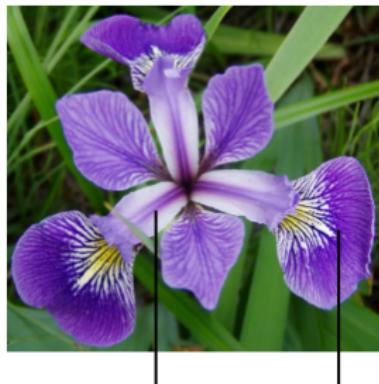
- *Application Phase:*

$$p_{\boldsymbol{\theta}^*}(y|\mathbf{x}_{\text{new}})$$

Classification



(a) Setosa ($y = 1$)



(b) Versicolor ($y = 2$)



(c) Virginica ($y = 3$)

Figure: Iris flower dataset ($\mathbf{x} \in \mathbb{R}^{256 \times 256 \times 3}$ and $y \in \{1, 2, 3\}$)

Classification

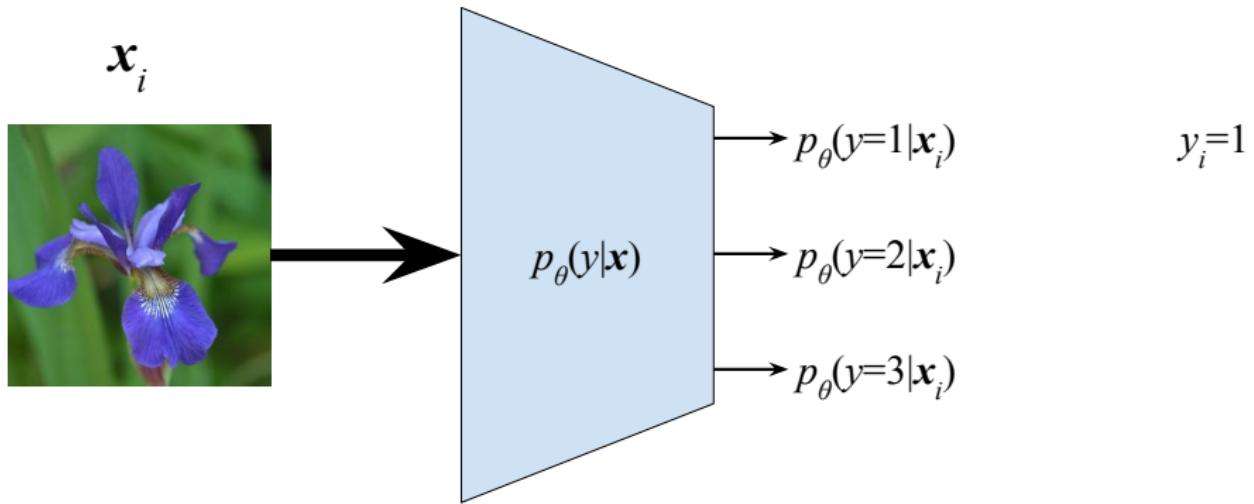


Figure: Training step 1: Evaluate $p_{\theta}(y|\mathbf{x}_i)$

Classification

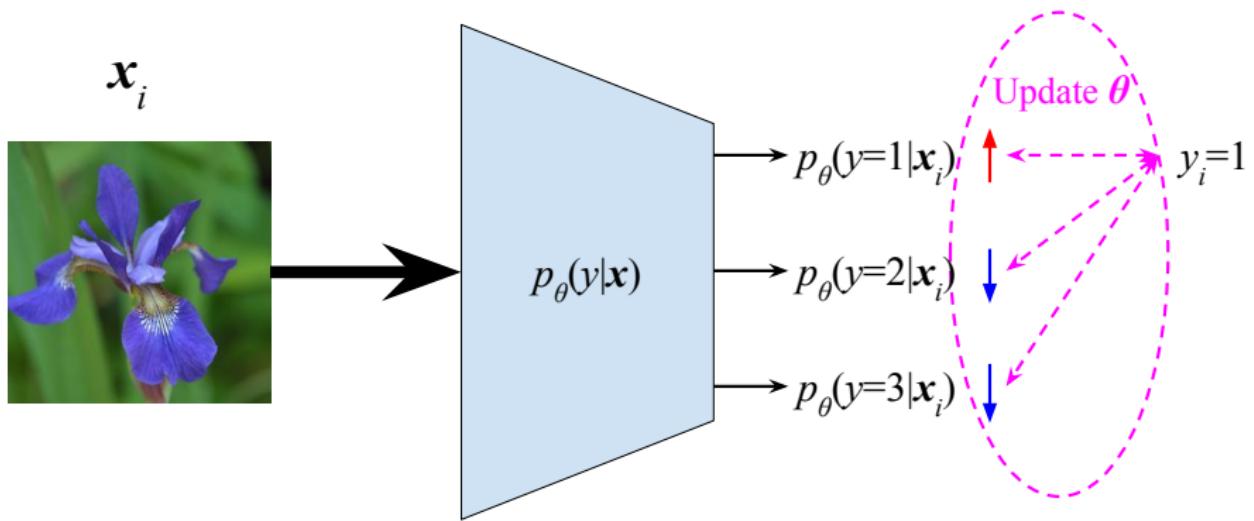


Figure: Training step 2: Update θ to minimize cross entropy

Classification

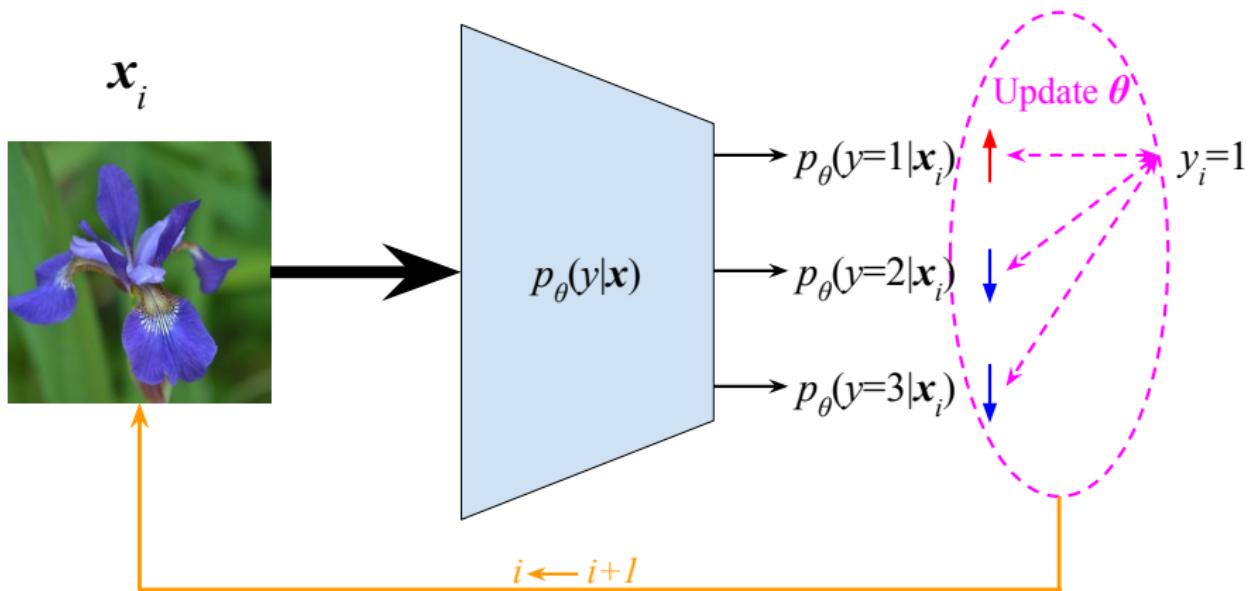


Figure: Training step 3: Go to the next training data

Classification

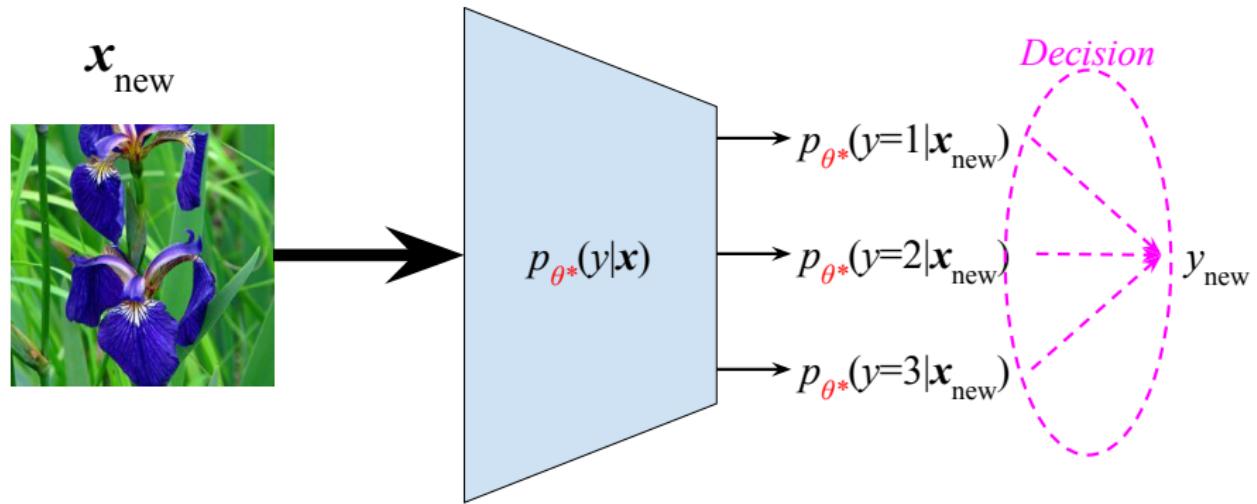


Figure: Inference: Evaluating the trained model θ^*

Multinomial Logistic Regression Model

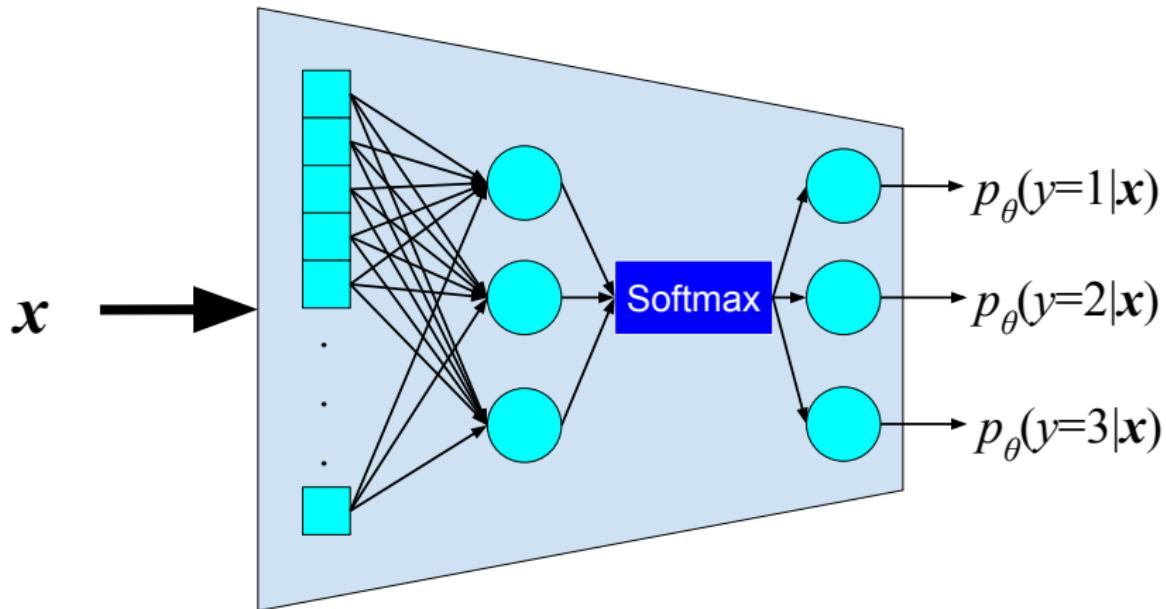


Figure: Multinomial logistic regression (MLR) model to implement $p_{\theta}(y|\mathbf{x})$ for Iris flower classification

Unsupervised Learning

Unsupervised Learning

- **Observations:** $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$

- **Applications:**

- Dimensionality reduction
- Clustering
- ...

- **Model:** Dimensionality Reduction $\Rightarrow \mathbf{y} = \mathbf{W}\mathbf{x}$,
$$\begin{cases} \mathbf{x} \in \mathbb{R}^D \\ \mathbf{W} \in \mathbb{R}^{L \times D} \\ \mathbf{y} \in \mathbb{R}^L \end{cases}$$

- **Phases:**

- *Training Phase:*

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^N \text{Loss}(\mathbf{x}_n, \mathbf{W}^T \mathbf{W} \mathbf{x})$$

- *Application Phase:*

$$\mathbf{y}_{\text{new}} = \mathbf{W}^* \mathbf{x}_{\text{new}}$$

Dimensionality Reduction

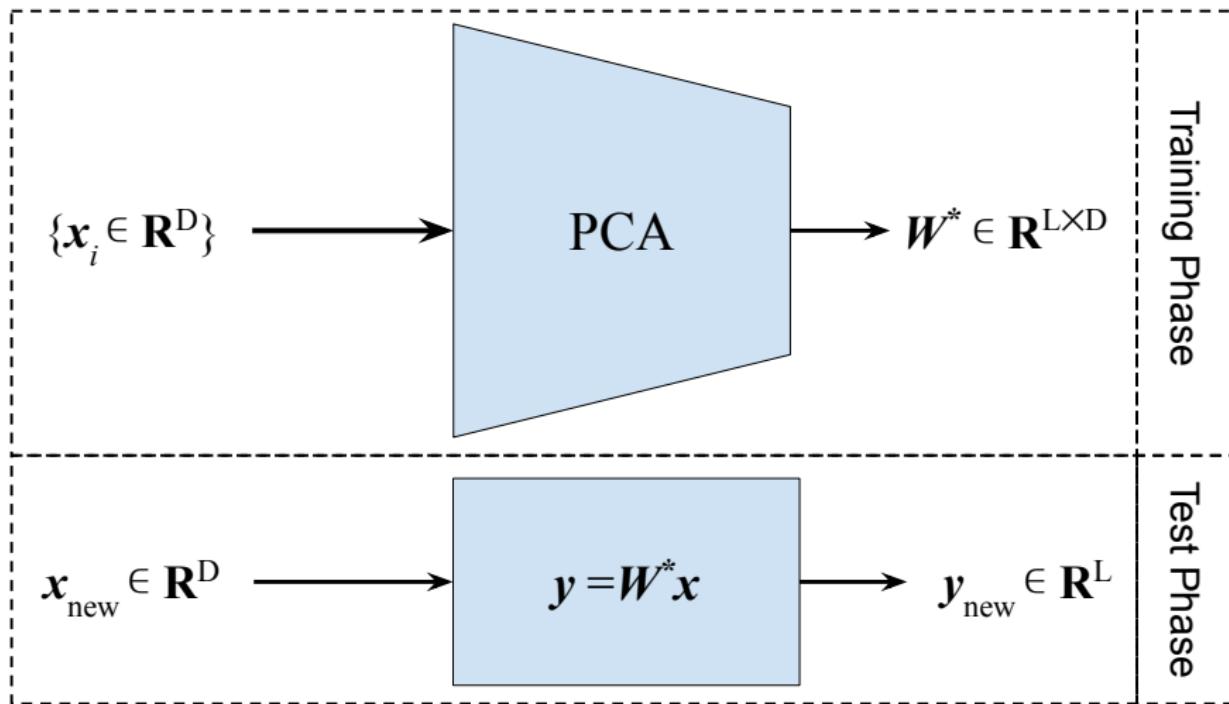


Figure: Training and Test for Dimensionality Reduction

Dimensionality Reduction

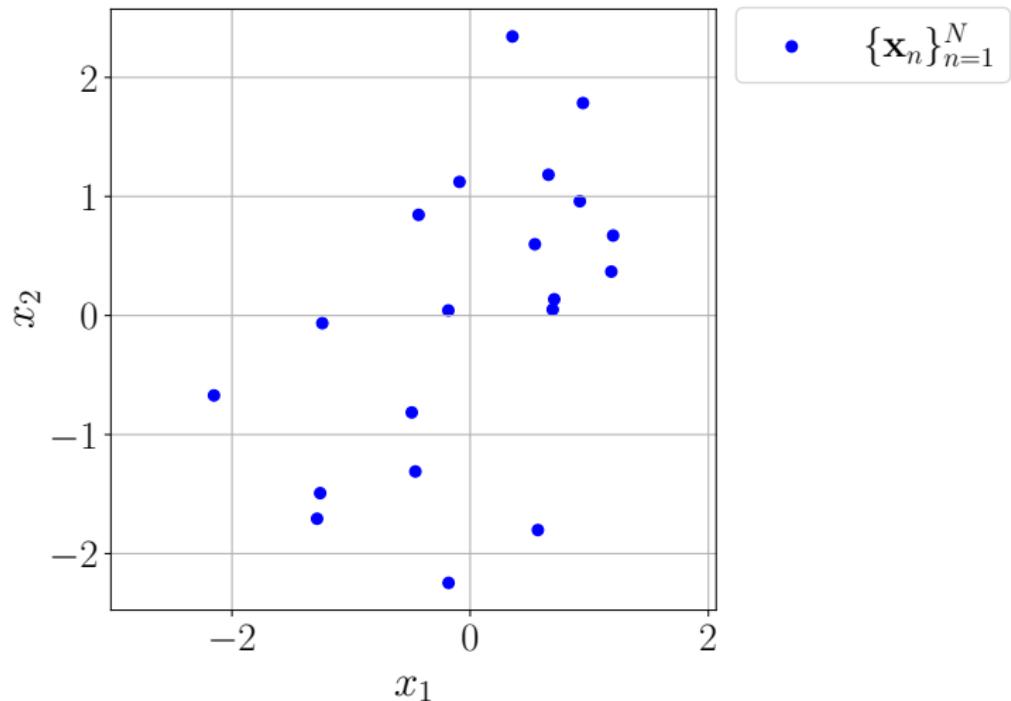


Figure: Training data for dimensionality reduction

Dimensionality Reduction

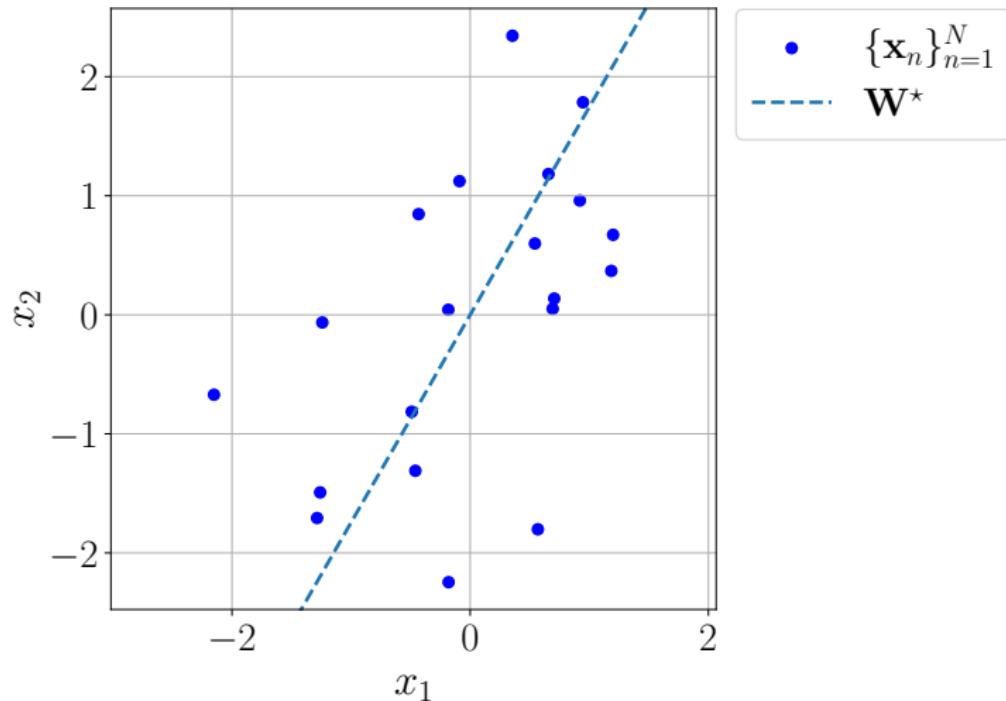


Figure: Learning principle component or projection matrix \mathbf{W}^*

Dimensionality Reduction

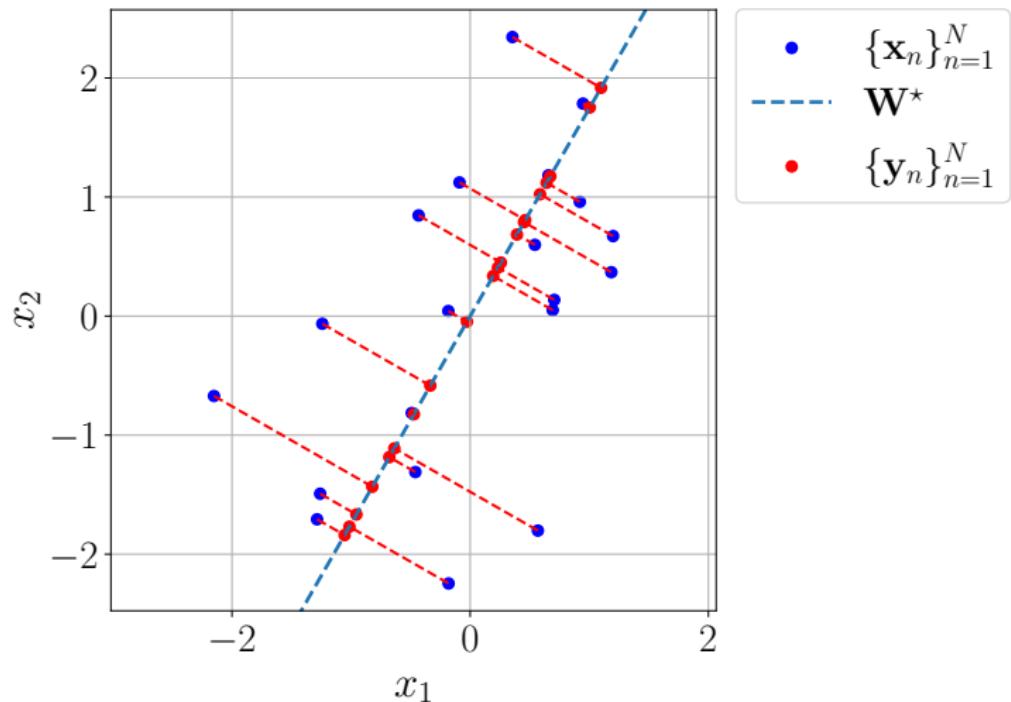


Figure: Reducing the dimensionality of training data

Altogether

Supervised Learning

- **Experience:**

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$$

- **Applications:**

- Classification
- Regression
- Semantic segmentation
- ...

- **Objective:**

- Estimating $p_{\text{data}}(y|\mathbf{x})$

Unsupervised Learning

- **Experience:**

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$$

- **Applications:**

- Clustering
- Dimensionality Reduction
- Density estimation
- ...

- **Objective:**

- Estimating $p_{\text{data}}(\mathbf{x})$

Section 2

Unconditional Generative Models
as Unsupervised Learning Models

Unconditional Generative Model

Generative Model Learning

- **Observation:**

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$$

- **Model:** $p_\theta(\mathbf{x})$

- **Tasks:**

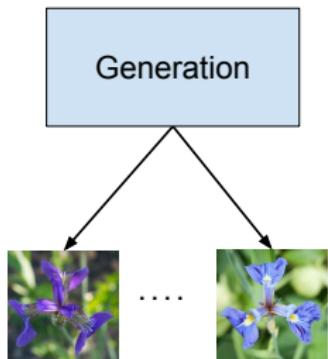
- Density estimation: $p_\theta(\mathbf{x}_{\text{new}})$
- Generation: $\mathbf{x}_{\text{new}} \sim p_\theta(\mathbf{x})$
- High level representation \mathbf{z} (unsupervised representation learning):
 - Assume, \mathbf{x} is IRIS flower, then ideally $\mathbf{z} \in \{1, 2, 3\}$ corresponding to each type.

- **Objective:**

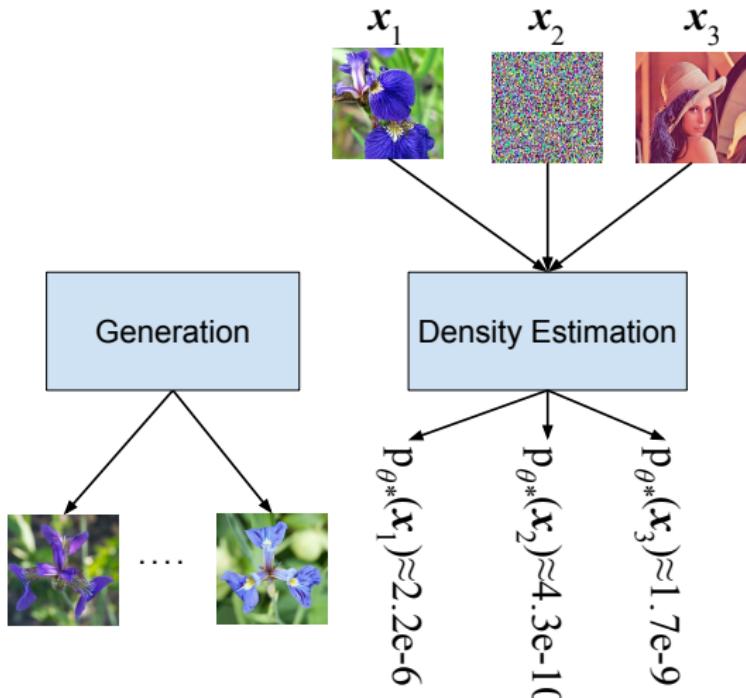
- Estimating distribution $p_{\text{data}}(\mathbf{x})$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \operatorname{dist}(p_{\text{data}}, p_{\boldsymbol{\theta}})$$

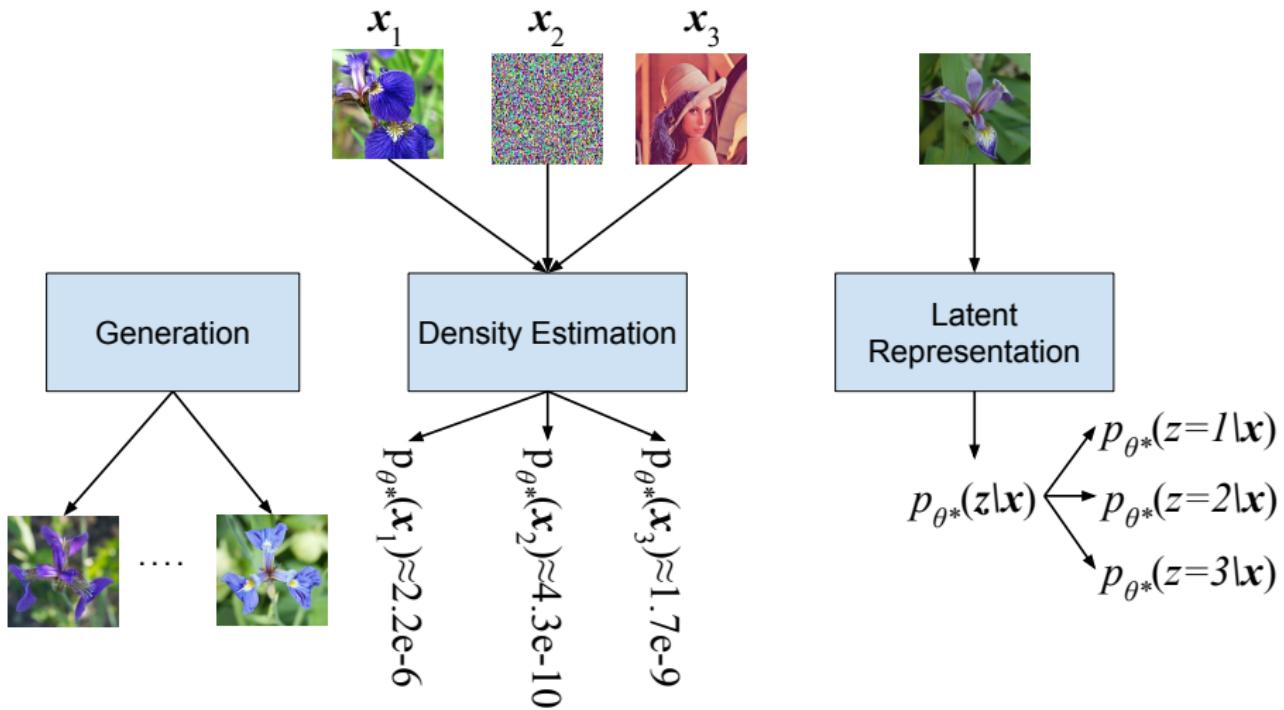
Unconditional Generative Model



Unconditional Generative Model



Unconditional Generative Model



Section 3

Conditional Generative Models
as Extension of Supervised Learning Models

Conditional Generative Model

Conditional Generative Model Learning

- **Observation:**

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$$

- **Model:** $p_\theta(\mathbf{y}|\mathbf{x})$

- **Tasks:**

- Conditional density estimation: $p_\theta(\mathbf{y}|\mathbf{x}_{\text{new}})$
- Generation: $\mathbf{y}_{\text{new}} \sim p_\theta(\mathbf{y}|\mathbf{x}_{\text{new}})$
- High level representation \mathbf{z} (unsupervised representation learning)

- **Objective:**

- Estimating distribution $p_{\text{data}}(\mathbf{y}|\mathbf{x})$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \operatorname{dist}(p_{\text{data}}, p_{\boldsymbol{\theta}})$$

Conditional Generative Model - Density Estimation

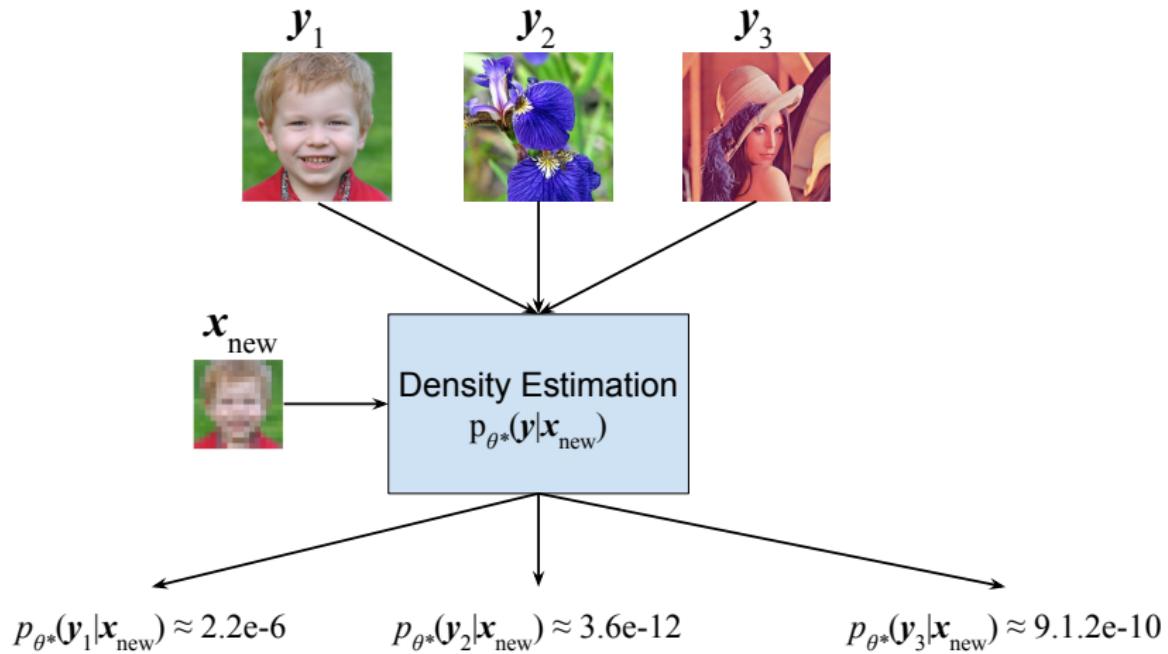


Figure: Assume we trained a conditional generative model and estimate θ^* . Then we can use it for density estimation of different y vector given x_{new} (image source: [1])

Conditional Generative Model - Generation

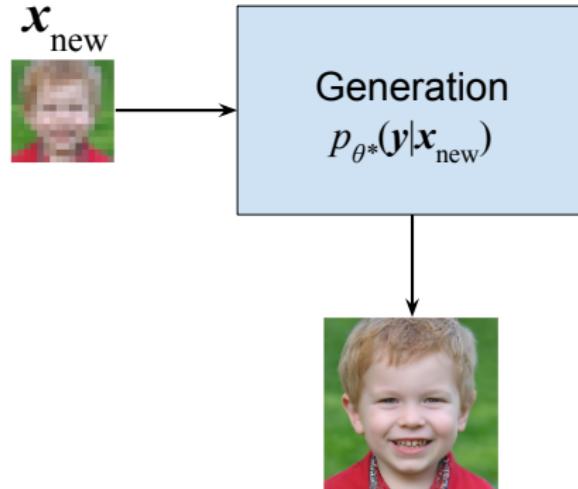


Figure: Assume we trained a conditional generative model and estimate θ^* . Then we can use it for sampling y given x_{new} (image source: [1])

Conditional Generative Model - Generation

MLR for Conditional Generative Model

Assume a simple super-resolution problem:

- $\mathbf{x} \in \{0, 1\}^{r \times c}$
- $\mathbf{y} \in \{0, 1\}^{h \times w}$

Then MLR must assign a probability to each possible output which is:

$$\text{Number of possible } \mathbf{y} \text{ vectors} \Rightarrow 2^{h \times w}$$

$$\text{Number of MLR parameters} \Rightarrow r \times c \times 2^{h \times w}$$

So using MLR and more complex models such as supervised deep models is not practical.

Conditional vs Unconditional Deep Generative Models

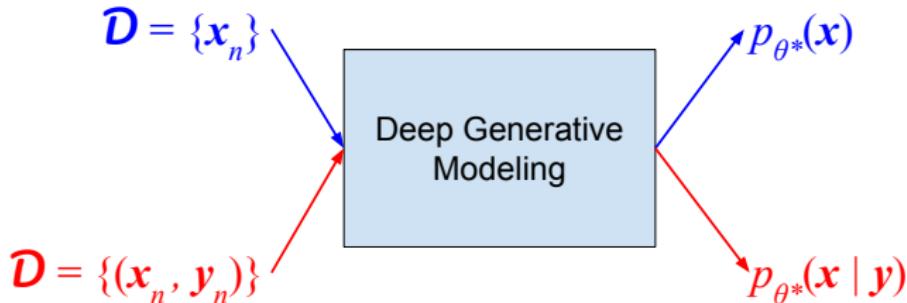


Figure: The methods to train conditional and unconditional generative modeling are almost shared in different approaches

Notation Update

To better share the notation across conditional and unconditional DGM, we use $p_{\theta}(x|y)$ instead of $p_{\theta}(y|x)$ for the conditional case. So we are always working on a probabilistic model over x .

Text-to-Speech Models

Text-to-Speech Models

$$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{An audio file} \\ \mathbf{y} : \text{A text} \end{cases}$$

Real-World Sample

Listen to the following speech synthesis (source: [2])

“A single Wavenet can
capture the characteristics of many
different speakers with equal fidelity,
not it’s fast.”

$\mathbf{y} =$ $\xrightarrow{\text{Sampling } p(\mathbf{x}|\mathbf{y})} \mathbf{x} =$

Text-to-Image Models

Text-to-Image Models

$$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{An image} \\ \mathbf{y} : \text{A text} \end{cases}$$



Figure: \mathbf{x} for \mathbf{y} = “Teddy bears swimming at the Olympics 400m Butterfly event.”
(source: [3])

Image-to-Image Translation

Image Colorization

$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{A Colored image} \\ \mathbf{y} : \text{A Gray - scale image} \end{cases}$



(a) \mathbf{y}



(b) \mathbf{x}



(c) Ground truth

Figure: Image colorization (source: [4])

Image-to-Image Translation

Image Inpainting

$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{A clean image} \\ \mathbf{y} : \text{A corrupted image} \end{cases}$



(a) \mathbf{y}



(b) \mathbf{x}



(c) Ground truth

Figure: Image inpainting (source: [4])

Image-to-Image Translation

Image Uncropping

$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{A clean image} \\ \mathbf{y} : \text{A cropped image} \end{cases}$



(a) \mathbf{y}



(b) \mathbf{x}



(c) Ground truth

Figure: Image uncropping (source: [4])

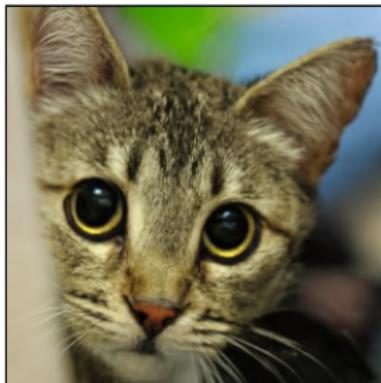
Image-to-Image Translation

Image Restoration

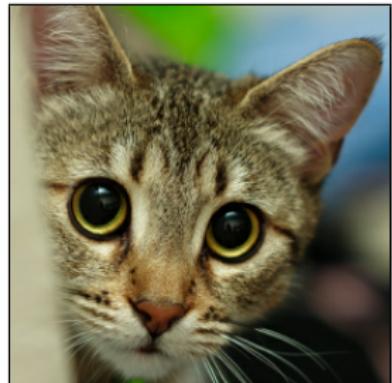
$$p(\mathbf{x}|\mathbf{y}) : \begin{cases} \mathbf{x} : \text{A clean image} \\ \mathbf{y} : \text{A degraded image} \end{cases}$$



(a) \mathbf{y}



(b) \mathbf{x}



(c) Ground truth

Figure: Image restoration (source: [4])

Section 4

Discriminative Vs. Generative Classification

Discriminative Approach for Classification

Multinomial Logistic Regression

- Model

$$p_{\theta}(y|\boldsymbol{x}) = \text{Cat}(y|\mathcal{S}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}))$$

where:

$y \in \{1, 2, 3, \dots, C\}$	Label
$\boldsymbol{x} \in \mathbb{R}^D$	Input pattern
$\mathcal{S}(\cdot)$	Softmax function
$\boldsymbol{W} \in \mathbb{R}^{C \times D}$	Weight matrix
$\boldsymbol{b} \in \mathbb{R}^C$	Bias vector
$\theta = (\boldsymbol{W}, \boldsymbol{b})$	Model parameters

- Number of parameters:

$$C \times D + C$$

Discriminative Approach for Classification

Multinomial Logistic Regression

Assume:

$$\mathbf{x} = \text{vec} \left(\begin{array}{c} \text{Image of a bedroom} \\ \in \mathbb{R}^{600 \times 800} \end{array} \right)$$

$$y \in \{1 \text{ (LivingRoom)}, 2 \text{ (BedRoom)}, 3 \text{ (DiningRoom)}\}$$

then:

$$p_{\theta} \left(y \middle| \begin{array}{c} \text{Image of a bedroom} \end{array} \right) = \text{Cat}(y | [0.1, 0.7, 0.2]^T) \xrightarrow{\text{Sampling}} \begin{cases} 2 \\ 2 \\ 2 \\ 3 \\ 2 \end{cases}$$

Generative Approach for Classification

Discriminant Analysis

- Model

$$p_{\theta}(y = c|\mathbf{x}) = \frac{p_{\theta}(\mathbf{x}|y = c)p_{\theta}(y = c)}{\sum_{c'} p_{\theta}(\mathbf{x}|y = c')p_{\theta}(y = c')}$$

where:

$$y \in \{1, 2, 3, \dots, C\}$$

Label

$$\mathbf{x} \in \mathbb{R}^D$$

Input pattern

$$p_{\theta}(y = c)$$

Prior probability over labels

$$p_{\theta}(\mathbf{x}|y = c)$$

Class conditional density

$$\boldsymbol{\theta}$$

Model parameters

- Number of Parameters (Assuming $p_{\theta}(\mathbf{x}|y = c)$ to be Multivariate Normal):

$$(C - 1) + \underbrace{C \times D}_{\text{Mean Vector}} + \underbrace{C \times \frac{D(D + 1)}{2}}_{\text{Covariance matrix}}$$

Generative Approach for Classification

Discriminant Analysis

Assume:

$$\mathbf{x} = \text{vec} \left(\begin{array}{c} \text{Image of a room} \\ \end{array} \right) \in \mathbb{R}^{600 \times 800}$$

$$y \in \{1 \text{ (LivingRoom)}, 2 \text{ (BedRoom)}, 3 \text{ (DiningRoom)}\}$$

then:

$$p_{\theta}(\mathbf{x}|y=2) \xrightarrow{\text{Sampling}} \left\{ \begin{array}{c} \text{Image 1} \\ \text{Image 2} \\ \text{Image 3} \\ \text{Image 4} \end{array} \right\}$$

Comparing Discriminative and Generative Approaches

Training

- Number of parameters (Assuming $p_{\theta}(\mathbf{x}|y = c)$ to be Multivariate Normal):

$$\overbrace{C \times D + C}^{\text{Discriminative}} \ll \underbrace{(C - 1) + C \times D + C \times \frac{D(D + 1)}{2}}_{\text{Generative}}$$

- Training a Discriminative model is much simpler

Challenging Situations

- Discriminative Model:

$$p_{\theta} \left(y \middle| \text{[Image of a person with a redacted face]} \right) = \text{Cat}(y | [0.7, 0.1, 0.2]^T)$$

- Generative Model: Assume \mathbf{x}^v and \mathbf{x}^m to be the visible and masked parts, respectively, we can calculate $p(y|\mathbf{x}^v)$ and thus:

$$\max_y p(y|\mathbf{x}^v)$$

and decide based on the visible part which is much more *robust*.

Section 5

Big Picture

Course Big Picture

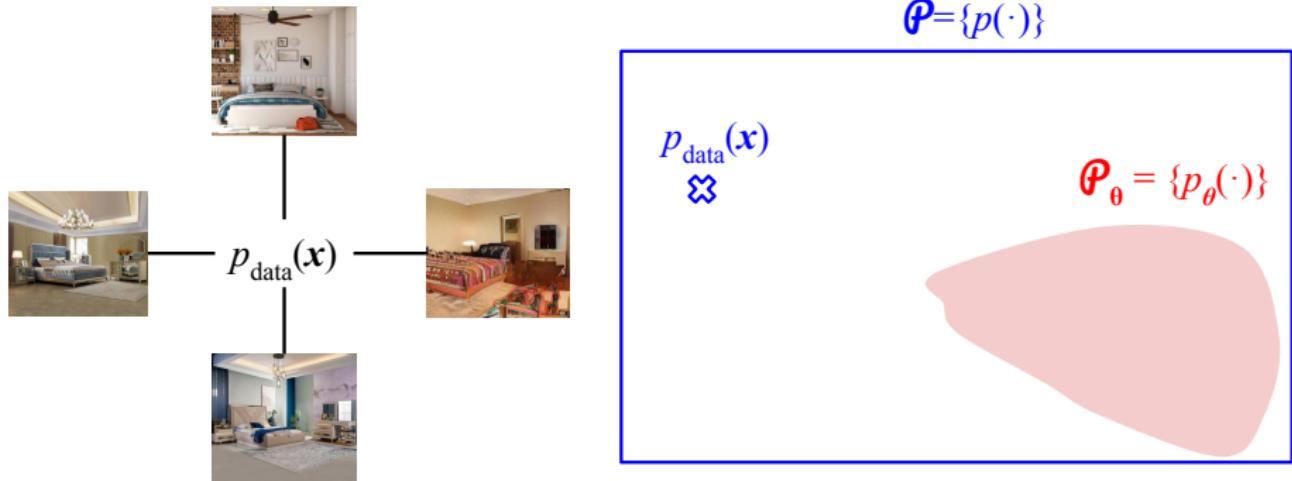


Figure: Selecting a family [5]

Course Big Picture

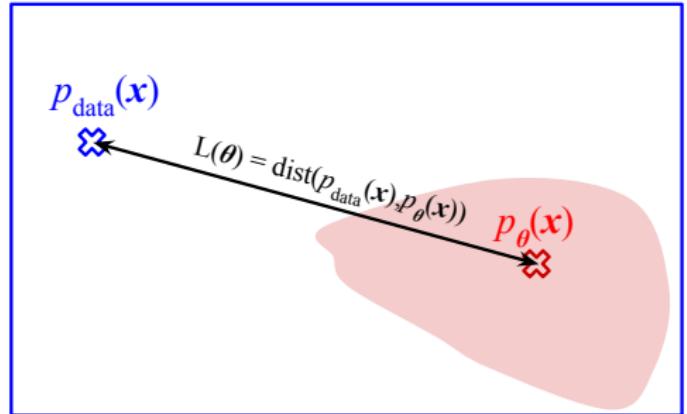
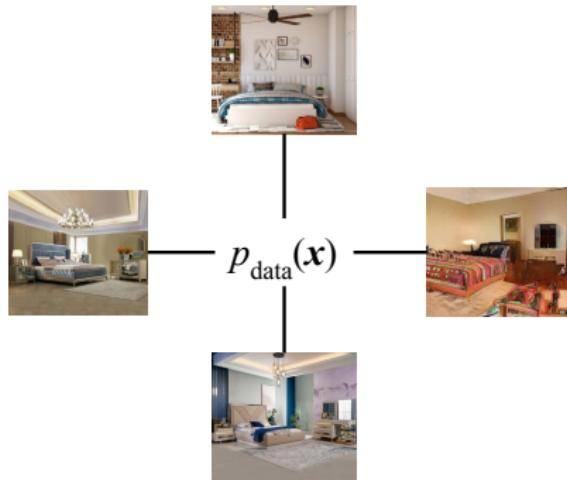
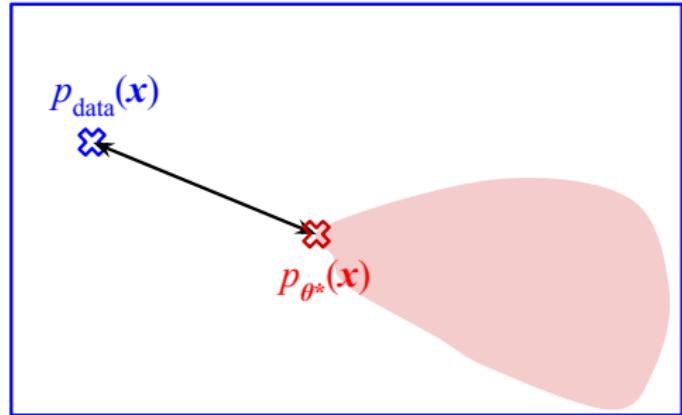
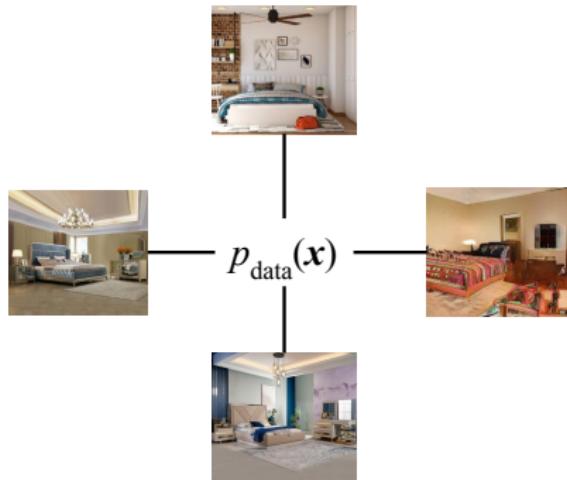


Figure: Developing a distance metric [5]

Course Big Picture



$$\theta^* = \operatorname{argmin}_{\theta} L(\theta)$$

Figure: Minimizing the distance metric [5]

Limited Vs. Unlimited Selection

$$\rho = \rho_{\theta}$$

$$p_{\text{data}}(x)$$

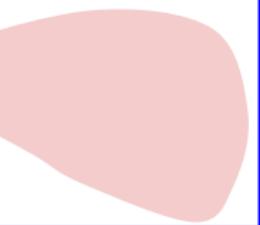
$$p_{\theta^*}(x)$$

$$\rho$$

$$p_{\text{data}}(x)$$


$$L(\theta^*)$$

$$p_{\theta^*}(x)$$

$$\rho_{\theta}$$


(a) Unlimited distribution selection

(b) Limited distribution selection

Generative Model for Simple Random Variables

Bernoulli Random Variable (Tossing a coin)

Assume we have a Bernoulli random variable:

$$X \sim \text{Ber}(\beta) \Rightarrow \begin{cases} p(X = 1) = \beta \\ p(X = 0) = 1 - \beta \end{cases}$$

Then:

- The description for all possible distributions is:

$$\mathcal{P} = \{\text{Ber}(\alpha); 0 \leq \alpha \leq 1\} \Rightarrow \# \text{Parameters} = 1$$

- The description for a subset of the whole family is:

$$\mathcal{P}_\theta = \{\text{Ber}(\theta); 0.2 \leq \theta \leq 0.3\} \Rightarrow \# \text{Parameters} = 1$$

Generation and Density Estimation from a Trained Model

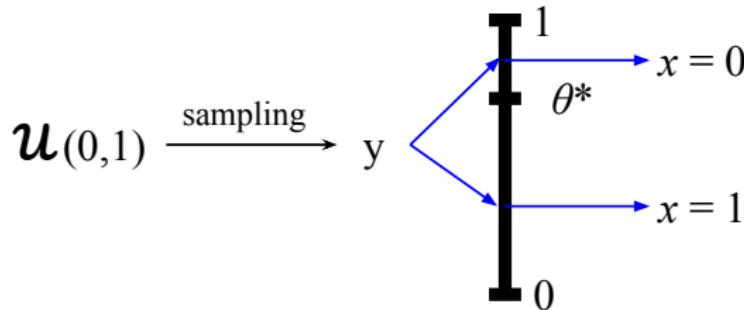


Figure: Sample generation using a trained model for Bernoulli distribution

Density Estimation

You have two states in your probabilistic world $X \in \{0, 1\}$, thus:

$$\begin{cases} p(X = 1) = \theta^* \\ p(X = 0) = 1 - \theta^* \end{cases}$$

Generative Model for Simple Random Variables

Categorical Random Variable (Rolling a die)

Assume we have a Categorical random variable with six states:

$$X \sim \text{Cat}([\beta_1, \dots, \beta_6]) \Rightarrow p(X = i) = \beta_i$$

Then:

- The description for all possible distributions is:

$$\mathcal{P} = \{\text{Cat}(\boldsymbol{\alpha}); 0 \leq \alpha_i \leq 1, \sum_i \alpha_i = 1\} \Rightarrow \#\text{Parameters} = 6 - 1$$

- The description for a subset of the whole family is:

$$\begin{aligned} \mathcal{P}_{\boldsymbol{\theta}} &= \left\{ \text{Cat}(\boldsymbol{\theta}); 0 \leq \theta_i \leq 1, \sum_i \theta_i = 1, \begin{cases} \theta_1 = \theta_2 = \theta_3 \\ \theta_4 = \theta_5 = \theta_6 \end{cases} \right\} \\ &\Rightarrow \#\text{Parameters} = 2 - 1 \end{aligned}$$

Generation and Density Estimation from a Trained Model

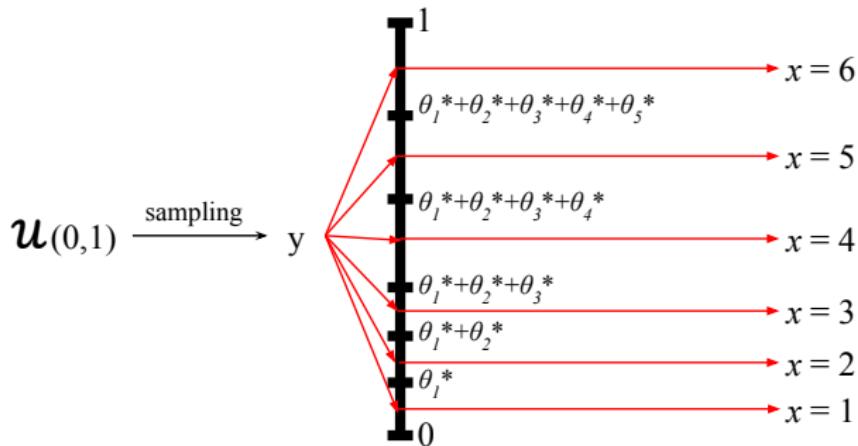


Figure: Sample generation using a trained model for Bernoulli distribution

Density Estimation

You have six states in your probabilistic world $X \in \{1, 2, 3, 4, 5, 6\}$, thus:

$$p(X = i) = \theta_i^*$$

Real-World Categorical Distribution

RGB pixel

Assume $(r, g, b) \sim p(R, G, B)$, $\begin{cases} \text{SampleSpace}(R) = \{0, \dots, 255\} \\ \text{SampleSpace}(G) = \{0, \dots, 255\} \\ \text{SampleSpace}(B) = \{0, \dots, 255\} \end{cases}$ to be an outcome of distributions over RGB pixel, then:

- Number of Categories: 256^3
- #Parameters for \mathcal{P} : $256^3 - 1$

Real-World Binary Image Distribution

Binary image with n pixels

Assume $\mathbf{x} \sim p(\mathbb{X})$ to be a binary image of digits where $\mathbb{X} = [X_1, \dots, X_n]$, Then:

- Number of Categories: 2^n
- #Parameters for \mathcal{P} : $2^n - 1$



(a) Generated samples from p_{θ^*} assuming $\mathcal{P} = \mathcal{P}_{\theta}$ we can find p_{θ^*} efficiently (source: [5])

$$\begin{array}{c} p_{\text{data}}(x) \\ \bowtie \\ p_{\theta^*}(x) \end{array}$$

Independence as Simplifying Tool

Binary image with n independent pixels

Assume $\mathbf{x} \sim p(\mathbb{X})$ to be a binary image of digits where the pixels are independent random variables, in other words, $p(\mathbf{x}) = p(x_1) \times p(x_2) \dots p(x_n)$, Then:

- Number of Categories: 2^n
- #Parameters for \mathcal{P} : $2^n - 1$
- #Parameters for \mathcal{P}_θ : n



(a) Generated samples from p_{θ^*} assuming we can find p_{θ^*} efficiently (source: [5])

Chain Rule

Binary image with n pixels

Using chain rule, we can write:

$$p(\mathbf{x}) = \overbrace{p(x_1)}^{\#1} \overbrace{p(x_2|x_1)}^{\#2} \dots \overbrace{p(x_n|x_{n-1}, \dots, x_1)}^{\#2^{n-1}}$$

Then:

- Number of Categories: 2^n
- #Parameters for \mathcal{P} : $2^0 + 2^1 + 2^2 + \dots + 2^{n-1} = 2^n - 1$

Chair rule

Note that the chain rule does not impose any restriction on the distribution. Thus again we are parameterizing the \mathcal{P} and thus the number of needed parameters is unchanged.

Chain Rule + Conditional Independence

Binary Image with Conditional Independence Property

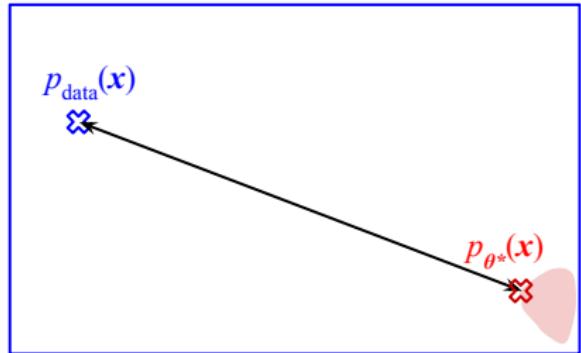
Now assume $X_{i+1} \perp X_{i-1}, \dots, X_1 | X_i$, then we can simplify the distribution as:

$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)\dots p(x_n|x_{n-1}, \dots, x_1) \\ &= \underbrace{p(x_1)}_{\#1} \underbrace{p(x_2|x_1)}_{\#2} \underbrace{p(x_3|x_2)}_{\#2} \dots \underbrace{p(x_n|x_{n-1})}_{\#2} \end{aligned}$$

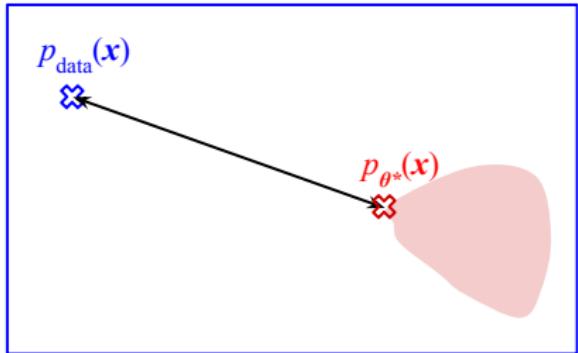
Then:

- Number of Categories: 2^n
- #Parameters for \mathcal{P} : $2n - 1$

Chain Rule + Conditional Independence



(a) Typical \mathcal{P} and \mathcal{P}_θ for independent case



(b) Typical \mathcal{P} and \mathcal{P}_θ for first-order Markov chain

Figure: With first-order Markov chain, while controlling the tractability for the number of parameters, the chance to find a better fit to p_{data} increases

Section 6

Conclusions

Conclusions

Our Objective in This Course

- Introduce suitable subset of probability distribution functions \mathcal{P}_θ :
 - Not too large (proposing \mathcal{P} as \mathcal{P}_θ)
 - Not too small (image with independent pixels)
 - Aware of input type (such as image, text, audio, video, etc.)
- Introduce suitable distance metrics between two distributions
- Introduce the optimization methods to reduce the distance metric between model distribution and true distribution
- Evaluate your modeling performance

List of Abbreviations

Complete	Abbreviation
Deep Generative Model	DGM
Multinomial Logistic Regression	MLR

References I

-  Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi,
“Image super-resolution via iterative refinement,”
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 4713–4726, 2022.
-  Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu,
“Wavenet: A generative model for raw audio,”
arXiv preprint arXiv:1609.03499, 2016.
-  Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al.,
“Photorealistic text-to-image diffusion models with deep language understanding,”
Advances in Neural Information Processing Systems, vol. 35, pp. 36479–36494, 2022.
-  Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi,
“Palette: Image-to-image diffusion models,”
in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.
-  Stefano Ermon and Yang Song,
“Cs236: Deep generative models,” Fall 2021.