

Deep Generative Models Background

Fall Semester 2025

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Outline

- **Basics of Probability, Statistics, Information Theory**
 - Discrete and Continuous Distributions, Independence
 - Marginals, Conditionals, Product Rule, Bayes Rule & Examples for Gaussians
 - Expectations, Covariance, Entropy, KL Divergence, Mutual Information
- Generative vs Discriminative Models
- Learning Generative Models
 - Learning Criterion: Maximum Likelihood Estimation
 - Learning Algorithm: Stochastic Gradient Descent
- Classes of Generative Models
 - Gaussian Models: Closed form Solution
 - General Models: Need for Structure
 - Taxonomy of Models
 - Latent variable models, Autoregressive models, Energy based models

Review of Probability and Statistics

- Data $\mathbf{x} \in \mathcal{X}$ follows some data distribution $\mathbf{x} \sim p_{\theta}(\mathbf{x})$ with parameter θ .
 - Properties: $p_{\theta}(\mathbf{x}) \geq 0$ (non-negativity), $\int p_{\theta}(\mathbf{x})d\mathbf{x} = 1$ (add up to 1)
- **Continuous case:** $\mathbf{x} \in \mathbb{R}^D$, $p_{\theta}(\mathbf{x})$ is a probability density function.
 - E.g., Gaussian distribution: $\mathbf{x} \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\mu} \in \mathbb{R}^D$, $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$, $\boldsymbol{\Sigma} \succcurlyeq 0$

$$p_{\theta}(\mathbf{x}) = (2\pi)^{-\frac{D}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

- **Discrete case:** $\mathbf{x} \in \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$, $p_{\theta}(\mathbf{x})$ is a probability mass function.
 - E.g., Categorical distribution: $\mathbf{x} \sim \text{Cat}(\mathbf{x} \mid \boldsymbol{\pi})$, $\theta = \boldsymbol{\pi}$, $\pi_k \geq 0$, $\sum_k \pi_k = 1$

$$p_{\theta}(\mathbf{x} = \mathbf{x}_k) = \pi_k$$

- **Independence:** \mathbf{x} and \mathbf{y} are independent if and only if $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$.

Marginals, Conditionals, Product Rule, Bayes Rule

- **Marginal distribution**

- Continuous case:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

- Discrete case:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$

- **Conditional distribution**

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

- **Product rule**

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y}) = p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x})$$

- **Bayes rule**

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

Example: Marginal and Conditional for a Gaussian

- Assume $\mathbf{x} \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_a & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ab}^\top & \boldsymbol{\Sigma}_b \end{bmatrix}$$

- Then, we get the following results

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a \mid \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a),$$
$$p(\mathbf{x}_a \mid \mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a \mid \hat{\boldsymbol{\mu}}_a, \hat{\boldsymbol{\Sigma}}_a),$$

where

$$\hat{\boldsymbol{\mu}}_a = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_b^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b)$$
$$\hat{\boldsymbol{\Sigma}}_a = \boldsymbol{\Sigma}_a - \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Sigma}_{ab}^\top$$

Warm-up exercise -> HW1

Example: Marginal for a Mixture of Gaussians

- Assume $\mathbf{y} \sim \text{Cat}(\mathbf{y} \mid \boldsymbol{\pi})$, $\mathbf{y} \in \{1, \dots, K\}$.
- Assume $\mathbf{x} \mid \mathbf{y} \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$.
- Then, \mathbf{x} is a mixture of Gaussians

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) &= \sum_{\mathbf{y}} p(\mathbf{x} \mid \mathbf{y}) p(\mathbf{y}) &= \sum_{k=1}^K p(\mathbf{x} \mid \mathbf{y} = k) p(\mathbf{y} = k) \\ \text{Marginalization} && \text{Product rule} && = \sum_{k=1}^K \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \pi_k \end{aligned}$$

Expectations and Covariance

- **Expectation**

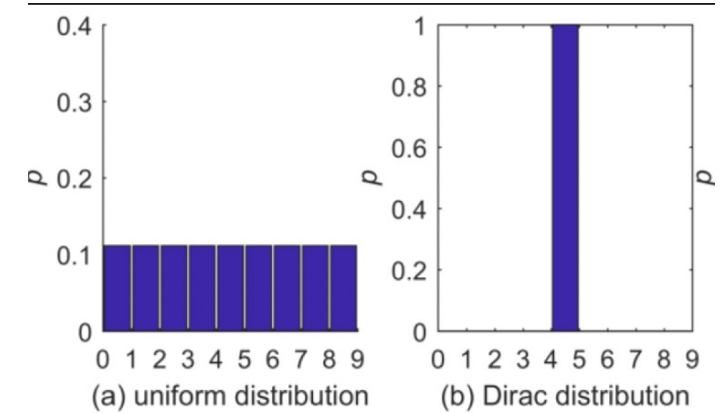
- Continuous case: $\boldsymbol{\mu}_x = \mathbb{E}[\boldsymbol{x}] = \int \boldsymbol{x} p(\boldsymbol{x}) d\boldsymbol{x}$
- Discrete case: $\boldsymbol{\mu}_x = \mathbb{E}[\boldsymbol{x}] = \sum_k \boldsymbol{x}_k p(\boldsymbol{x} = \boldsymbol{x}_k)$

- **Covariance**

- Continuous case:
 - $\boldsymbol{\Sigma}_x = \mathbb{V}[\boldsymbol{x}] = \int (\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^\top p(\boldsymbol{x}) d\boldsymbol{x}$
 - $\boldsymbol{\Sigma}_{xy} = \text{Cov}[\boldsymbol{x}, \boldsymbol{y}] = \int (\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{y} - \boldsymbol{\mu}_y)^\top p(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}$
- Discrete case:
 - $\boldsymbol{\Sigma}_x = \mathbb{V}[\boldsymbol{x}] = \sum_k (\boldsymbol{x}_k - \boldsymbol{\mu}_x)(\boldsymbol{x}_k - \boldsymbol{\mu}_x)^\top p(\boldsymbol{x} = \boldsymbol{x}_k)$
 - $\boldsymbol{\Sigma}_{xy} = \text{Cov}[\boldsymbol{x}, \boldsymbol{y}] = \sum_k (\boldsymbol{x}_k - \boldsymbol{\mu}_x)(\boldsymbol{y}_k - \boldsymbol{\mu}_y)^\top p(\boldsymbol{x} = \boldsymbol{x}_k, \boldsymbol{y} = \boldsymbol{y}_k)$

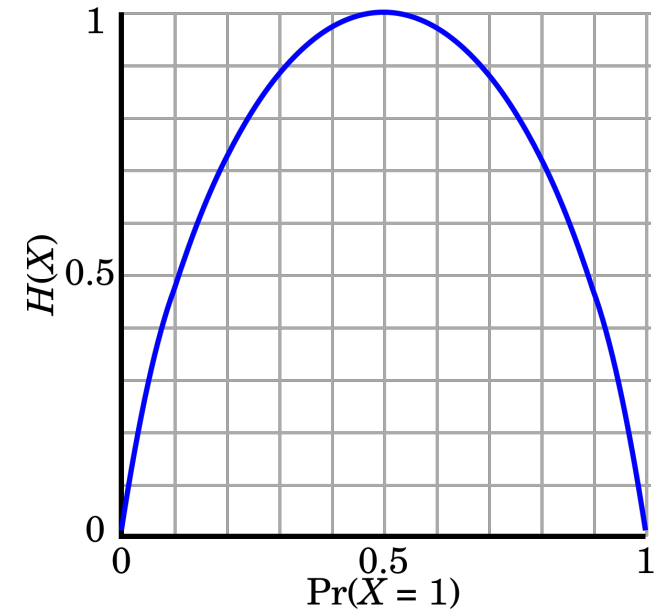
Entropy and Conditional Entropy

- **Entropy** of a random variable \mathbf{x}
 - It captures how much “uncertainty” is present in \mathbf{x} .
 - **Definition:** $H(\mathbf{x}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[-\log p(\mathbf{x})]$
 - **Continuous:** $H(\mathbf{x}) = -\int_{\mathbf{x}} \log(p(\mathbf{x})) p(\mathbf{x}) d\mathbf{x}$
 - **Discrete:** $H(\mathbf{x}) = -\sum_k \log(\pi_k) \pi_k$ where $\pi_k = P(\mathbf{x} = k)$
- **Examples:**
 - **Uniform:** $\pi_k = \frac{1}{K} \Rightarrow H(\mathbf{x}) = -\sum_k \log(\frac{1}{K}) \frac{1}{K} = \log(K)$
 - **Bernoulli:** $\pi_1 = q \Rightarrow H(\mathbf{x}) = -q \log q - (1 - q) \log(1 - q)$
- **Conditional entropy:** uncertainty of \mathbf{x} when \mathbf{y} is observed
 - $H(\mathbf{x} \mid \mathbf{y}) = \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p(\mathbf{x}, \mathbf{y})}[-\log p(\mathbf{x} \mid \mathbf{y})]$



High entropy

Low entropy



Entropy of a Bernoulli variable

Kullback–Leibler Divergence and Mutual Information

- **KL divergence** measures the similarity between two distributions p, q

$$\text{KL}[p \parallel q] = \mathbb{E}_{\mathbf{x} \sim p} \left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right]$$

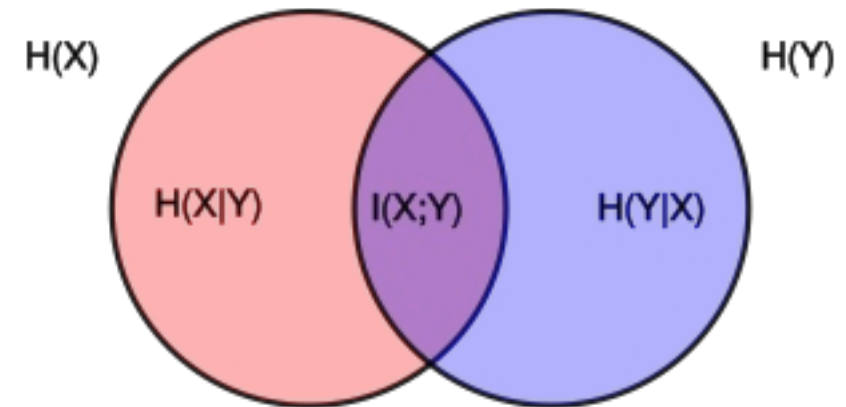
- Non-negativity $\text{KL}[p \parallel q] \geq 0$. Equality holds iff $p = q$.
- In general triangle inequality and symmetry does not hold.

- **Mutual Information** measures the mutual dependence between \mathbf{x} and \mathbf{y}

$$I(\mathbf{x}; \mathbf{y}) = \text{KL}[p(\mathbf{x}, \mathbf{y}) \parallel p(\mathbf{x})p(\mathbf{y})] = \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p(\mathbf{x}, \mathbf{y})} \left[\log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right) \right]$$

- If \mathbf{x}, \mathbf{y} are independent, then $I(\mathbf{x}; \mathbf{y}) = 0$.
- $I(\mathbf{x}; \mathbf{y})$ measures the uncertainty in \mathbf{x} after observing \mathbf{y} .

$$I(\mathbf{x}; \mathbf{y}) = H(\mathbf{x}) - H(\mathbf{x} \mid \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y} \mid \mathbf{x})$$

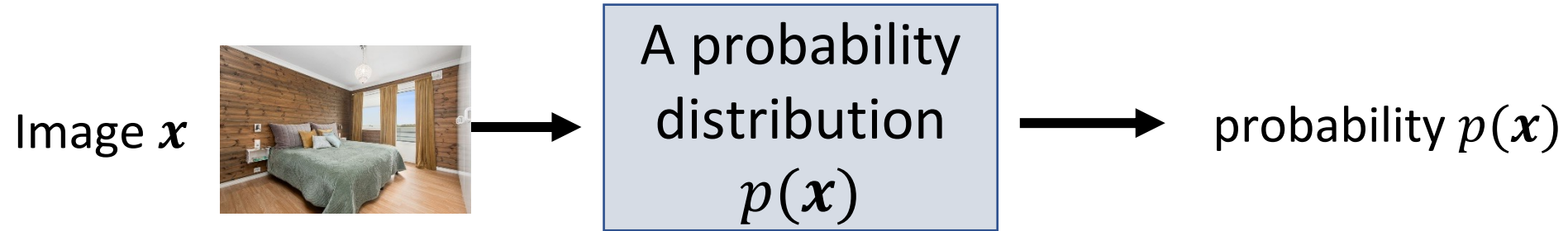


Outline

- Basics of Probability, Statistics, Information Theory
 - Discrete and Continuous Distributions, Independence
 - Marginals, Conditionals, Product Rule, Bayes Rule & Examples for Gaussians
 - Expectations, Covariance, Entropy, KL Divergence, Mutual Information
- **Generative vs Discriminative Models**
- Learning Generative Models
 - Learning Criterion: Maximum Likelihood Estimation
 - Learning Algorithm: Stochastic Gradient Descent
- Classes of Generative Models
 - Gaussian Models: Closed form Solution
 - General Models: Need for Structure
 - Taxonomy of Models
 - Latent variable models, Autoregressive models, Energy based models

Statistical Generative Models

- A statistical generative model is a probability distribution $p(\mathbf{x})$



- It is generative because **sampling from $p(\mathbf{x})$ generates new images**



...



Discriminative vs. Generative Models

Discriminative: classify bedroom vs. dining room

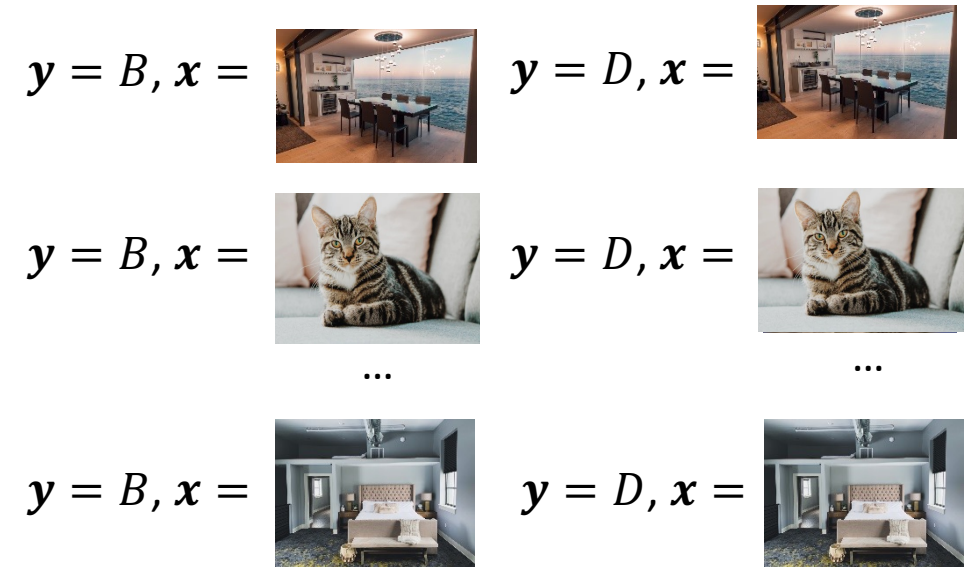


The image x is given. **Goal:** decision boundary, via **conditional distribution over label y**

$$P(y = \text{Bedroom} \mid x = \text{[Dining Room Image]}) = 0.0001$$

Ex: logistic regression, convolutional net, etc.

Generative: generate x






The input x is **not** given. Requires a model of the **joint distribution over both x and y**


$$P(y = \text{Bedroom}, x = \text{[Dining Room Image]}) = 0.0002$$

Discriminative vs. Generative

Joint and conditional are related via **Bayes Rule**:

$$P(\mathbf{y} = \text{Bedroom} \mid \mathbf{x} = \text{img1}) = \frac{P(\mathbf{y} = \text{Bedroom}, \mathbf{x} = \text{img2})}{P(\mathbf{x} = \text{img3})}$$




Discriminative: \mathbf{y} is simple; \mathbf{x} is always given, so not need to model $P(\mathbf{x} = \text{img4})$



Thus, a discriminative model cannot handle missing data $P(\mathbf{y} = \text{Bedroom} \mid \mathbf{x} = \text{img5})$

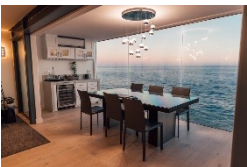


Conditional Generative Models


Class **conditional generative models** are also possible:

$$P(\mathbf{x} = \text{ | \mathbf{y} = \text{Bedroom})$$

It's often useful to condition on rich side information \mathbf{Y}

$$P(\mathbf{x} = \text{ | \text{Caption} = \text{"A black table with 6 chairs"})$$

A discriminative model is a very simple conditional generative model of \mathbf{y} :

$$P(\mathbf{y} = \text{Bedroom} | \mathbf{x} = \text{})$$

Why Generative Models?

- AI Is Not Only About Decision Making

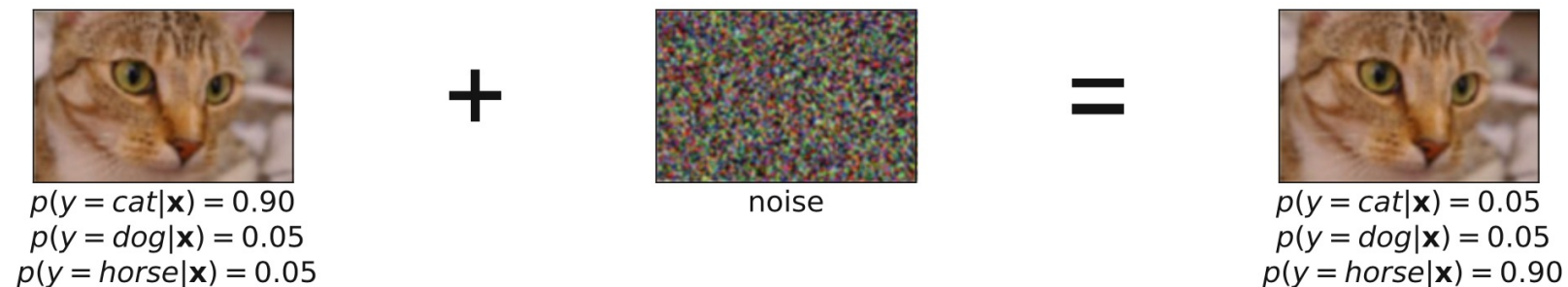


Fig. 1.1 An example of adding noise to an almost perfectly classified image that results in a shift of predicted label

- Importance of uncertainty and understanding in decision making

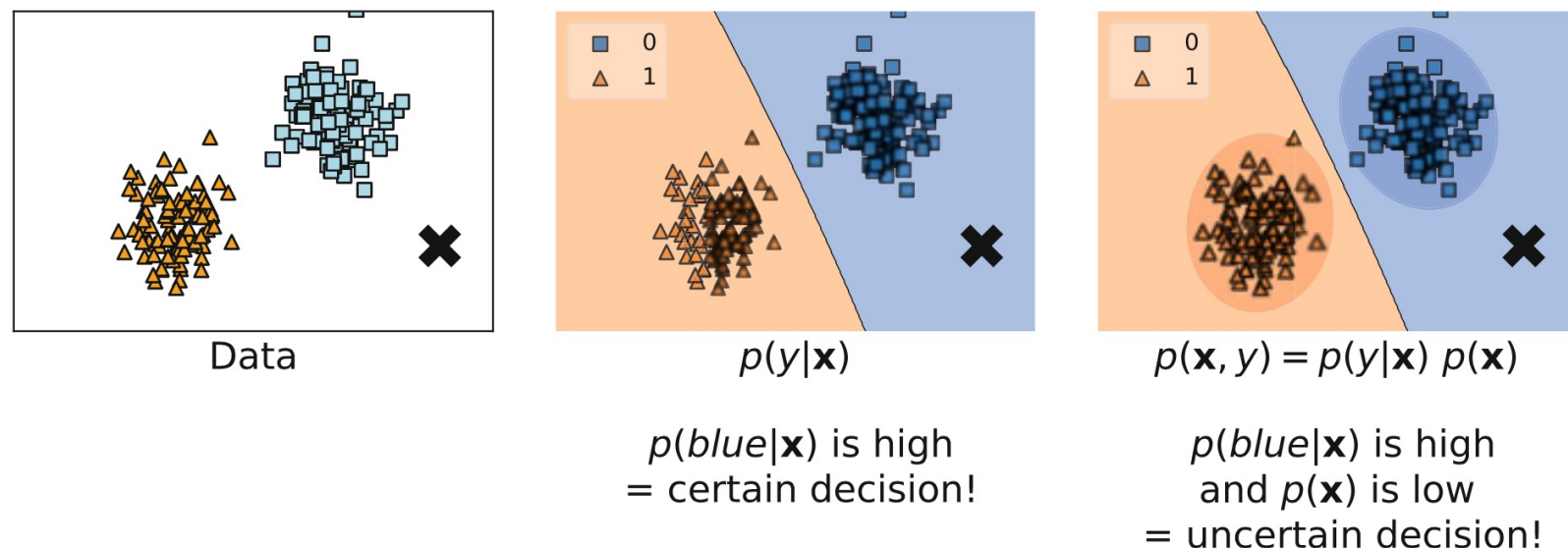


Fig. 1.2 An example of data (left) and two approaches to decision making: (middle) a discriminative approach and (right) a generative approach

Outline

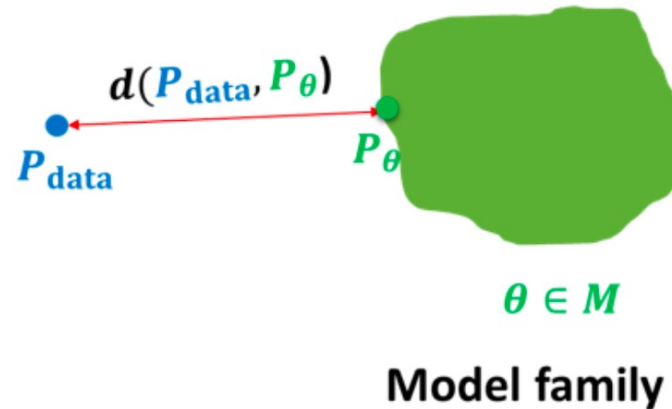
- Basics of Probability, Statistics, Information Theory
 - Discrete and Continuous Distributions, Independence
 - Marginals, Conditionals & Example for a Gaussian
 - Entropy, Mutual Information, KL Divergence
- Generative vs Discriminative Models
- **Learning Generative Models**
 - Learning Criterion: Maximum Likelihood Estimation
 - Learning Algorithm: Stochastic Gradient Descent
- Classes of Generative Models
 - Gaussian Models: Closed form Solution
 - General Models: Need for Structure
 - Taxonomy of Models
 - Latent variable models, Autoregressive models, Energy based models

Learning Generative Models

- We are given a training set of examples, e.g., images of dogs



$$x_i \sim P_{\text{data}} \\ i = 1, 2, \dots, n$$



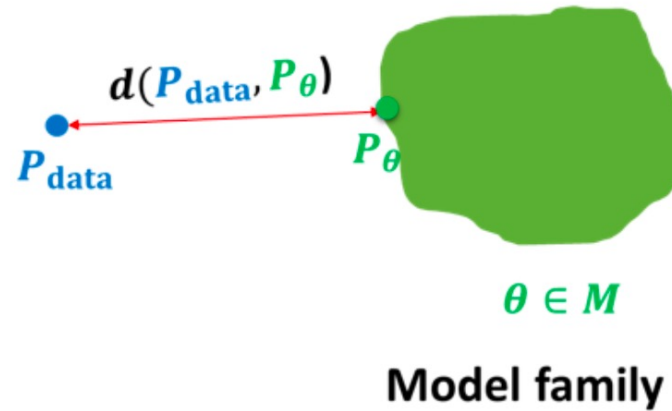
- We want to learn a probability distribution $p_{\theta}(x)$ over images x to allow for
 - **Generation:** If we sample $x_{\text{new}} \sim p_{\theta}(x)$, then x_{new} should look like a dog (sampling)
 - **Density estimation:** $p_{\theta}(x)$ should be high if x looks like a dog, and low otherwise (anomaly detection)
 - **Unsupervised representation learning:** We should be able to learn what these images have in common, e.g., ears, tail, etc. (features)

Learning Generative Models

- We are given a training set of examples, e.g., images of dogs



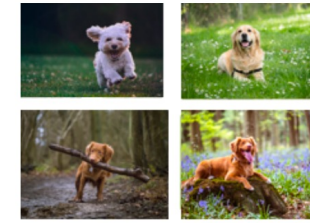
$$\begin{aligned} x_i &\sim P_{\text{data}} \\ i &= 1, 2, \dots, n \end{aligned}$$



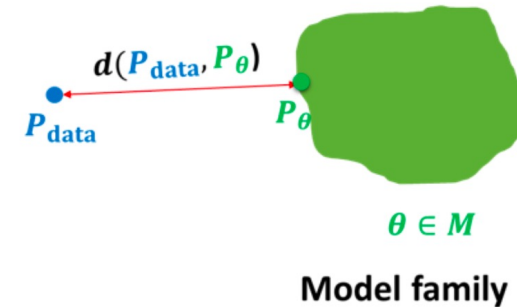
- We want to learn a probability distribution $p_{\theta}(x)$ over images x
- What learning criterion should we use?
- What optimization algorithm should we use?
- What classes of models should we learn?

Learning Criterion: Maximum Likelihood Estimation

- **Goal:** given a dataset $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of i.i.d. samples from the unknown data distribution $p_{\text{data}}(\mathbf{x})$, we want to learn a distribution $p_{\theta}(\mathbf{x})$ that is as “close” as possible to $p_{\text{data}}(\mathbf{x})$



$\mathbf{x}_i \sim P_{\text{data}}$
 $i = 1, 2, \dots, n$



- Taking d as the KL divergence introduced before: $\min_{\theta} KL[p_{\text{data}}(\mathbf{x}) || p_{\theta}(\mathbf{x})]$
- Since $KL[p_{\text{data}}(x) || p_{\theta}(x)] = E_{x \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(x)}{p_{\theta}(x)} \right]$ and we optimize over θ , the above problem is equivalent to $\max_{\theta} E_{x \sim p_{\text{data}}} [\log p_{\theta}(x)]$
- As we do not know the true distribution $p_{\text{data}}(x)$ and only have samples \mathcal{D} from it, we can replace the above objective with an unbiased estimate of it

$$\max_{\theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i)$$

This is the classic Maximum Likelihood Estimation (MLE) principle!

Maximum Likelihood Estimation (MLE)

- Likelihood is expressed as the joint distribution over all samples
- And by our i.i.d. assumption

$$\mathcal{L}(\theta) = p_{\theta}(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{i=1}^N p_{\theta}(\mathbf{x}_i)$$

- Taking the log, we can rewrite

$$\ell(\theta) = \log(\mathcal{L}(\theta)) = \log\left(\prod_{i=1}^N p_{\theta}(\mathbf{x}_i)\right) = \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i)$$

- The maximum likelihood estimator is the parameters that maximizes $\ell(\theta)$, i.e.

$$\hat{\theta}_{ML} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i)$$

Optimization Algorithm: Stochastic Gradient Descent

- **Goal:** optimize an objective that contains an expectation $\min_{\theta} g(\theta) := E_{x \sim p}[f(x, \theta)]$
- **Gradient descent:** first order algorithm to optimize $g(\theta)$
 - Tractable even when θ is in high dimensions
 - Gradient descent: $\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_{\theta} g(\theta^{(k)})$
 - Many variants to accelerate / deal with non-differentiability
- **Challenge:** It is difficult to compute $\nabla_{\theta} g(\theta)$ in closed form
 - $\nabla_{\theta} g(\theta) = \nabla_{\theta} E_{x \sim p}[f(x, \theta)] = E_{x \sim p}[\nabla_{\theta} f(x, \theta)]$
 - Often p is the true data distribution which we do not know; we have samples from p
 - Even if we knew p , computing the integral is potentially very complicated f is difficult
- **Solution:** Approximate $\nabla_{\theta} g(\theta)$ with samples
 - Let x_1, \dots, x_b be a batch of i.i.d. samples from p
 - $\frac{1}{b} \sum_i^b \nabla_{\theta} f(x_i, \theta)$ is an unbiased estimator of $\nabla_{\theta} g(\theta)$
 - Stochastic gradient descent: $\theta^{(k+1)} = \theta^{(k)} - \eta \frac{1}{b} \sum_i^b \nabla_{\theta} f(x_i, \theta)$

Outline

- Basics of Probability, Statistics, Information Theory
 - Discrete and Continuous Distributions, Independence
 - Marginals, Conditionals & Example for a Gaussian
 - Entropy, Mutual Information, KL Divergence
- Generative vs Discriminative Models
- Learning Generative Models
 - Learning Criterion: Maximum Likelihood Estimation
 - Learning Algorithm: Stochastic Gradient Descent
- **Classes of Generative Models**
 - Gaussian Models: Closed form Solution
 - General Models: Need for Structure
 - Taxonomy of Models
 - Latent variable models, Autoregressive models, Energy based models

Gaussian Parameter Estimation via MLE

- Given: N i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ from an unknown Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in \mathbb{R}^D
- Goal: use MLE to estimate the parameters $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of the Gaussian distribution
- Recall Gaussian density: $p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$
- This allows us to write down the likelihood function...

$$\mathcal{L}(\theta) = \prod_{i=1}^N p_{\theta}(\mathbf{x}_i) = \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right)}{(2\pi)^{\frac{ND}{2}} \det(\boldsymbol{\Sigma})^{\frac{N}{2}}}$$

- ... and the log of the likelihood

$$\begin{aligned} \ell(\theta) &= \sum_{i=1}^N -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log(\det(\boldsymbol{\Sigma})) - (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \\ &= -\frac{ND}{2} \log(2\pi) - \frac{N}{2} \log(\det(\boldsymbol{\Sigma})) - \frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \end{aligned}$$

Optimizing the Log Likelihood

- Reminder: Log-likelihood objective

$$\ell(\theta) = -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\log(\det(\mathbf{\Sigma})) - \frac{1}{2}\sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

- To find the optimal θ , we take the derivatives of $\ell(\theta)$ and set them to 0

$$\frac{\partial \ell(\theta)}{\partial \boldsymbol{\mu}} = 0, \quad \frac{\partial \ell(\theta)}{\partial \mathbf{\Sigma}} = 0$$

- Taking the derivative log-likelihood w.r.t. to the mean yields

$$\frac{\partial \ell(\theta)}{\partial \boldsymbol{\mu}} = \sum_{i=1}^N \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) = 0 \Rightarrow \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu}) = 0 \Rightarrow \hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$$

Optimizing the Log Likelihood

- Reminder: Log-likelihood objective

$$\ell(\theta) = -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\log(\det(\mathbf{\Sigma})) - \frac{1}{2}\sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

- Before taking the derivative, we make a change of variables to handle the inverse covariance $\mathbf{S} = \mathbf{\Sigma}^{-1}$ and note the following identity involving traces

$$\mathbf{x}^\top \mathbf{S} \mathbf{x} = \text{tr}(\mathbf{x}^\top \mathbf{S} \mathbf{x}) = \text{tr}(\mathbf{S} \mathbf{x} \mathbf{x}^\top)$$

- We can then rewrite the log-likelihood in terms of \mathbf{S}

$$\ell(\theta) = -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\log \det(\mathbf{S}^{-1}) - \frac{1}{2}\text{tr}\left(\mathbf{S} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\right)$$

Optimizing the Log Likelihood

- From our re-written log-likelihood function

$$\ell(\theta) = -\frac{ND}{2}\log(2\pi) + \frac{N}{2}\log(\det(\mathbf{S})) - \frac{1}{2}\text{tr}\left(\mathbf{S}\sum_{i=1}^N(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\right)$$

- Taking the derivative with respect to \mathbf{S}

$$\frac{\partial \ell(\theta)}{\partial \mathbf{S}} = \frac{N}{2}\mathbf{S}^{-1} - \frac{1}{2}\sum_{i=1}^N(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top = 0$$

- Arriving at our desired ML estimator for the covariance

$$\hat{\boldsymbol{\Sigma}}_{ML} = \mathbf{S}^{-1} = \frac{1}{N}\sum_{i=1}^N(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

ML Estimators for Mean and Variance of a Gaussian

- **In summary:** Given N i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ from an unknown Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in \mathbb{R}^D
- The log-likelihood function is given by

$$\ell(\theta) = -\frac{ND}{2} \log(2\pi) - \frac{N}{2} \log(\det(\boldsymbol{\Sigma})) - \frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

- The maximum log likelihood estimators for the mean and covariance are

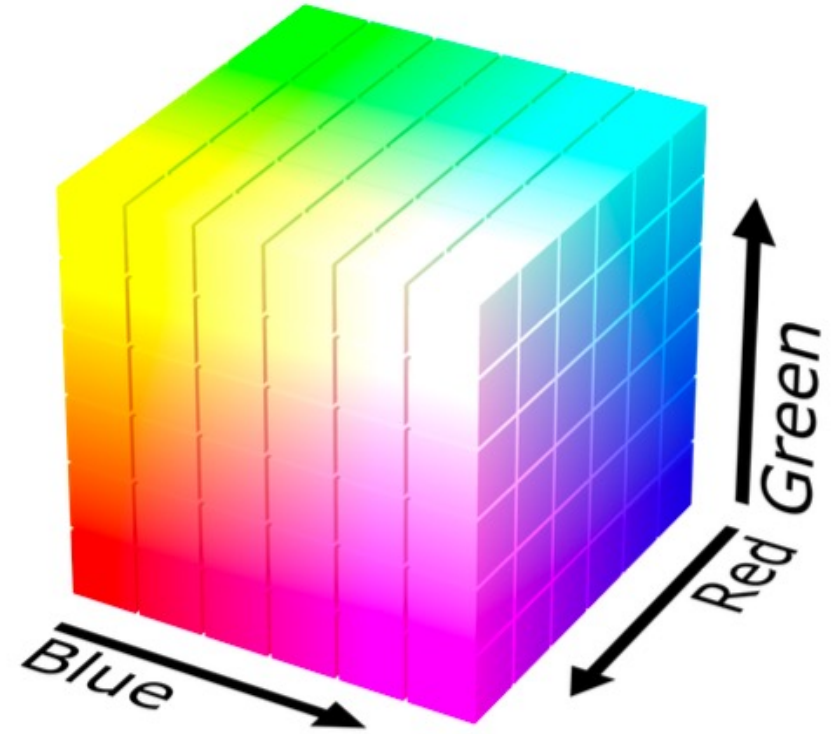
$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad \hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

Outline

- Basics of Probability, Statistics, Information Theory
 - Discrete and Continuous Distributions, Independence
 - Marginals, Conditionals & Example for a Gaussian
 - Entropy, Mutual Information, KL Divergence
- Generative vs Discriminative Models
- Learning Generative Models
 - Learning Criterion: Maximum Likelihood Estimation
 - Learning Algorithm: Stochastic Gradient Descent
- Classes of Generative Models
 - Gaussian Models: Closed form Solution
 - **General Models: Need for Structure**
 - Taxonomy of Models
 - Latent variable models, Autoregressive models, Energy based models

Example: RGB images

- To model a single pixel's color, one needs three discrete random variables:
 - Red Channel R taking values in $\{0, \dots, 255\}$
 - Green Channel G taking values in $\{0, \dots, 255\}$
 - Blue Channel B taking values in $\{0, \dots, 255\}$
- Sampling from the joint distribution $(r, g, b) \sim p(R, G, B)$ randomly generates a color for the pixel. How many parameters do we need to specify the joint distribution $p(R = r, G = g, B = b)$?



$$256 * 256 * 256 - 1$$

Example: Joint Distribution



- Suppose X_1, \dots, X_n are Bernoulli random variables modelling n pixels of an image
- How many possible states?

$$\underbrace{2 \times 2 \times \dots \times 2}_{n \text{ times}} = 2^n$$

- Sampling from $p(x_1, \dots, x_n)$ generates an image
- How many parameters to specify the joint distribution $p(x_1, \dots, x_n)$ over n binary pixels?

$$2^n - 1$$

Structure Through Independence

- If X_1, \dots, X_n are independent, then
$$p(x_1, \dots, x_n) = p(x_1)p(x_2) \cdots p(x_n)$$
- How many possible states? 2^n
- How many parameters to specify the joint distribution $p(x_1, \dots, x_n)$?
 - How many to specify the marginal distribution $p(x_1)$? 1
- 2^n entries can be described by just n numbers (if each X_i just take 2 values)!
- Independence assumption is too strong. Model not likely to be useful
 - For example, each pixel chosen independently when we sample from it.



Structure Through Conditional Independence

- Using Chain Rule

$$p(x_1, \dots, x_n) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \dots, x_{n-1})$$

- How many parameters? $1 + 2 + \dots + 2^{n-1} = 2^n - 1$
 - $p(x_1)$ requires 1 parameter
 - $p(x_2 \mid x_1 = 0)$ and $p(x_2 \mid x_1 = 1)$ requires 1 parameter each \Rightarrow 2 parameters in total
 - $p(x_n \mid x_1, \dots, x_{n-1})$ requires 2^{n-1} parameters
 - $2^n - 1$ is still exponential, chain rule does not buy us anything.

- Now suppose the **Markov property** holds, i.e., $X_{i+1} \perp X_1, \dots, X_{i-1} \mid X_i$, then

$$\begin{aligned} p(x_1, \dots, x_n) &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid \cancel{x_1}, x_2) \cdots p(x_n \mid \cancel{x_1, \dots}, x_{n-1}) \\ &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1}) \end{aligned}$$

- How many parameters? $2n - 1$. Exponential reduction!

Taxonomy of Generative Models

- To effectively learn generative models we need to impose some structure

- Autoregressive Models

$$p(\mathbf{x}) = p(x_0) \prod_{i=1}^D p(x_i | \mathbf{x}_{<i}),$$

- Latent Variable Models

$$\begin{aligned} \mathbf{z} &\sim p(\mathbf{z}) \\ \mathbf{x} &\sim p(\mathbf{x} | \mathbf{z}) \end{aligned}$$

- Energy Based Models

$$p(\mathbf{x}) = \frac{\exp\{-E(\mathbf{x})\}}{Z}$$

Taxonomy of Generative Models

