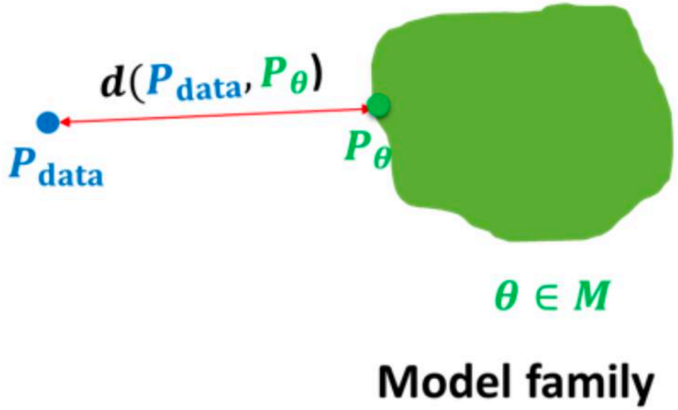
Outline

- 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

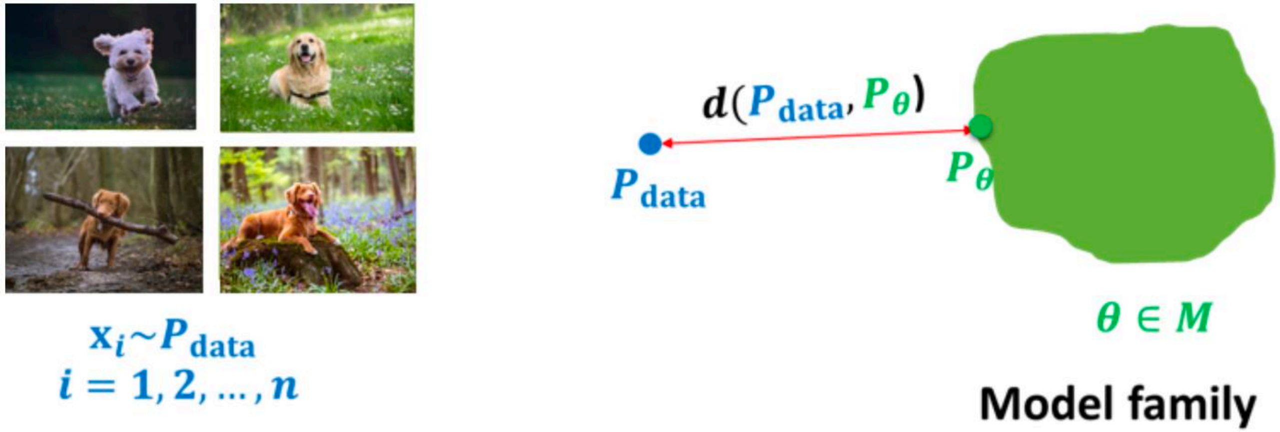
$\mathbf{x}_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



- We want to learn a probability distribution $p(x)$ over images x to allow for
  - **Generation**: If we sample $x_{\text{new}} \sim p(x)$, x_{new} should look like a dog (sampling)
  - **Density estimation**: $p(x)$ 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

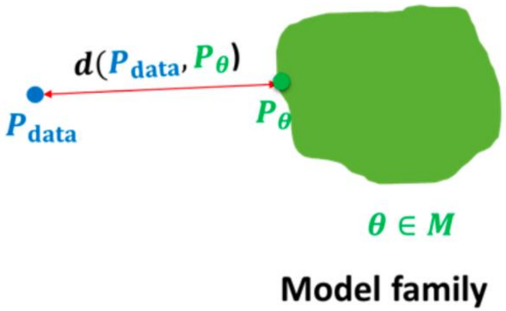


- What learning criterion should we use?
- What optimization algorithm should we use?
- What classes of models should we learn?

Learning Criterion: Maximum Likelihood Estimation

- Given: a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of i.i.d. samples from the unknown data distribution $p_{\text{data}}(x)$

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



- Goal: learn a distribution $p_\theta(x)$ parameterized by θ that is as close to $p_{\text{data}}(x)$
- Taking d as the KL divergence introduced before: $\min_\theta KL[p_{\text{data}}(x) \,||\, p_\theta(x)]$
- 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

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

- Taking the log, we can rewrite

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

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

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

Optimization Algorithm: Stochastic Gradient Descent

- Goal: optimize an objective that contains an expectation
$$\min_\theta g(\theta) \coloneqq E_{x\sim p}[f(x,\theta)]$$
- First order algorithms to optimize $g(\theta)$
  - Tractable even when θ is in high dimensions
  - Gradient descent: $\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_\theta g\left(\theta^{(k)}\right)$
  - 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 know p, integrating a potentially very complicated f is difficult
- Solution: Approximating $\nabla_\theta g(\theta)$ with samples
  - Let $x_1, \ldots, x_b$ 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 $x_1, \dots x_N$ from an unknown Gaussian $\mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^D$
- Goal: use MLE to estimate the parameters $\theta = (\mu, \Sigma)$ of the Gaussian distribution
- Recall Gaussian density: $p(x) = \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-\mathbf{1}}(\boldsymbol{x}-\boldsymbol{\mu})\right)$
- This allows us to write down the likelihood function…

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

- … and the log of the likelihood

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

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

Finding the gradient of parameters

- Reminder: Log-likelihood objective

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

- To find the optimal θ_{ML} , we take the derivatives of our objective w.r.t our parameters and set them to 0

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

For the mean

- Reminder: Log-likelihood objective

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

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

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

$$\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}) = 0$$

- Hence,

$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$$

For the covariance

- Reminder: Log-likelihood objective

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

- Before we find the derivative, we find a change of variable to handle the inverse covariance (also known as the precision matrix

$$\boldsymbol{S} = \boldsymbol{\Sigma}^{-1}$$

- And note the following identity involving traces

$$\boldsymbol{S}\boldsymbol{x} = \mathrm{tr}(\boldsymbol{x}^\top\boldsymbol{S}\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top)$$

For the covariance

- Reminder: Log-likelihood objective

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

- The two facts:
  - $\boldsymbol{S} = \boldsymbol{\Sigma}^{-1}$
  - $\boldsymbol{S}\boldsymbol{x} = \mathrm{tr}(\boldsymbol{x}^\top\boldsymbol{S}\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top)$
- Using these two facts, we can rewrite the log-likelihood in terms of $\boldsymbol{S}$ (omitting terms that derivative will cancel)

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

For the covariance

- From our re-written log-likelihood function

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

- Taking the derivative with respect to $\boldsymbol{S}$

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

- Arriving at our desired ML estimator for the covariance

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

ML Estimators for mean and variance

- The complete statement:

- If we assume our data samples are i.i.d Gaussians, the maximum log likelihood estimators for the mean and covariance are

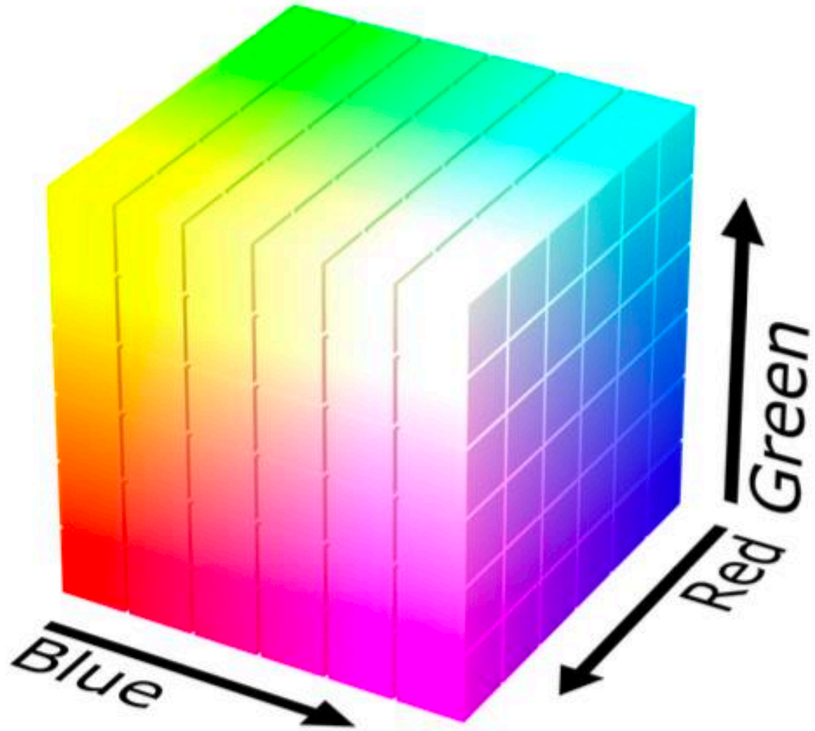
$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \qquad \hat{\boldsymbol{\Sigma}}_{ML} = \boldsymbol{S}^{-1} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$

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 modeling a single pixel's color, one needs three discrete random variables:
  - Red Channel R taking values in $\{0, \cdots, 255\}$
  - Green Channel G taking values in $\{0, \cdots, 255\}$
  - Blue Channel B taking values in $\{0, \cdots, 255\}$



- 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?

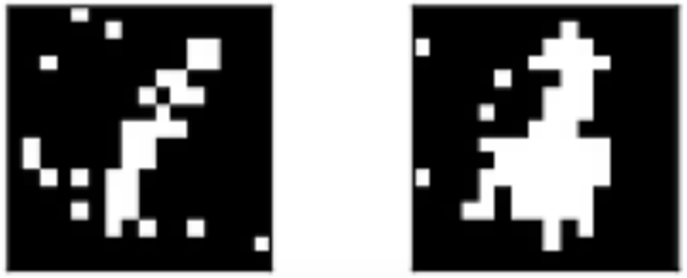
$$\underbrace{2 \times 2 \times \cdots \times 2}_{n \text{ times}} = 2^n$$

- 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

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

- How many parameters? $1 + 2 + \cdots + 2^{n-1} = 2^n - 1$
  - $p(x_1)$ requires 1 parameter
  - $p(x_2 \mid x_1 = 0)$ requires 1 parameter, $p(x_2 \mid x_1 = 1)$ requires 1 parameter Total 2 parameters.
  - . .
- $2^n - 1$ is still exponential, chain rule does not buy us anything.
- Now suppose $X_{i+1} \perp X_1, \ldots, X_{i-1} \mid X_i$, then

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

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

Taxonomy of Generative Models

- Autoregressive Models

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

- Latent Variable Models

$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

- Energy Based Models

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

Taxonomy of Generative Models