Deep Generative Models Background

Fall Semester 2024

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

Background of the course

- Basics of Probability, Statistics, Information Theory
  - Discrete and Continuous Distributions, Independence
  - Marginals, Conditionals
    - Example of a Gaussian
  - Entropy, Mutual Information, KL Divergence
- Basics of Optimization: Stochastic Gradient Descent
- Generative vs Discriminative models
- Need for structure
- Taxonomy of Generative Models
- Maximum Likelihood Estimation
- Latent variable models
  - Parameter estimation: Variational Inference and Expectation Maximization

Review of Probability and Statistics

- We define some basic notations
- Data $\boldsymbol{x} \in \mathbb{R}^D$ follows some data distribution $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- If $\boldsymbol{x}$ is discrete, then $p(\boldsymbol{x})$ is a probability mass function, taking on discrete values $k \in \mathcal{X} = \{1, ..., N\}$
- If $\boldsymbol{x}$ is continuous, then $p(\boldsymbol{x})$ is a probability density function
- Independence: $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent if and only if $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$

Marginals, Conditionals

- Marginal distribution
  - In the continuous case

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

  - In the discrete case

$$p(x) = \sum_y p(x, y)$$

- Conditional distribution

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

- Product rule

$$\begin{aligned} p(x, y) &= p(x|y)p(y) \\ &= p(y|x)p(x) \end{aligned}$$

- Bayes rule

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

Marginal and Conditional Distribution for a Gaussian

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

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_a & \boldsymbol{\Sigma}_c \\ \boldsymbol{\Sigma}_c^T & \boldsymbol{\Sigma}_b \end{bmatrix},$$

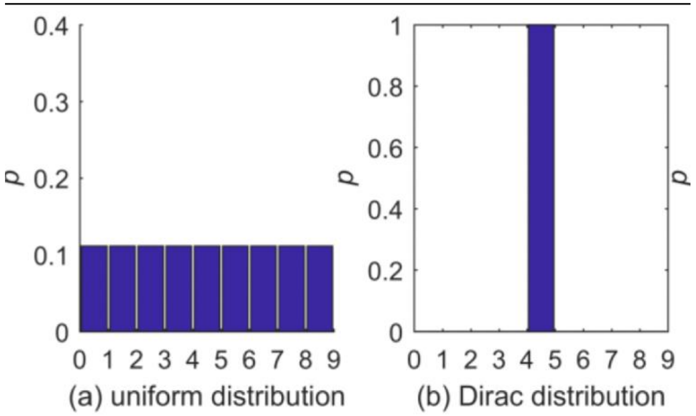
then we get the following dependencies:

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a),$$
$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\hat{\boldsymbol{\mu}}_a, \hat{\boldsymbol{\Sigma}}_a), \text{ where}$$
$$\hat{\boldsymbol{\mu}}_a = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_c\boldsymbol{\Sigma}_b^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b),$$
$$\hat{\boldsymbol{\Sigma}}_a = \boldsymbol{\Sigma}_a - \boldsymbol{\Sigma}_c\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\Sigma}_c^T.$$

Warm-up exercise -> HW1

Review of Information Theory

- **Entropy** of a random variable X captures how much "uncertainty" is present in X
  - Discrete case: $H(X) = H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log p_i$
  - Continuous case: $H(X) = -\int_x p(x) \log p(x) dx$



High entropy　　Low entropy

- **Mutual Information:**
  - mutual dependence between X and Y
  - reduction of uncertainty in X when Y is observed

$$I(X;Y) = H(X) - H(X \mid Y) = \mathrm{H(Y)} - \mathrm{H(Y \mid X)}$$

Conditional entropy: uncertainty of random variable conditioning on another variable

- Discrete case: $I(X;Y) = \sum_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$
- Continuous case: $I(X;Y) = \int_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$

Review of Information Theory

- **KL divergence** between two distributions p, q captures how similar p, q are
  - In the continuous case

$$KL[p(x)\ ||\ q(x)] = \int p(x) \log \frac{p(x)}{q(x)} dx$$

  - In the discrete case

$$KL[p(x)\ ||\ q(x)] = \sum p(x) \log \frac{p(x)}{q(x)}$$

  - Properties
    - Non-negativity $KL[p(x)\ ||\ q(x)] \geq 0$. Equality holds iff $p = q$
    - In general triangle inequality and symmetry does not hold

Review of Optimization: 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
  - $\nabla_\theta g(\theta) = \nabla_\theta E_{x\sim p}[f(x,\theta)] = E_{x\sim p}[\nabla_\theta f(x,\theta)]$
  - This involves doing the integral (expectation)
- Solution: Approximating $\nabla_\theta g(\theta)$ with samples
  - Let $x_1, \dots, x_n$ be i.i.d. samples from p
  - $\frac{1}{n}\sum_i^n \nabla_\theta f(x_i,\theta)$ is an unbiased estimator of $\nabla_\theta g(\theta)$
  - $\theta^{(k+1)} = \theta^{(k)} - \eta \frac{1}{n}\sum_i^n \nabla_\theta f(x_i,\theta)$

Statistical Generative Models

A statistical generative model is a **probability distribution** p(x)

- **Data:** samples (e.g., images of bedrooms)
- **Prior knowledge:** parametric form (e.g., Gaussian?), loss function (e.g., maximum likelihood?), optimization algorithm, etc.

Image x  → A probability distribution p(x) → probability p(x)

It is generative because **sampling from p(x) generates new images**

 … 

Discriminative vs. Generative

Discriminative: classify bedroom vs. dining room

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

P(Y = Bedroom | X=  0.0001

Ex: logistic regression, convolutional net, etc.

Generative: generate X

Y=B , X=  Y=D , X= 

Y=B , X=  Y=D , X= 

... ...

Y=B , X=  Y=D , X= 

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

P(Y = Bedroom , X= ) = 0.0002

Discriminative vs. Generative

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

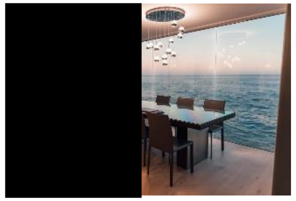
P(Y = Bedroom | X= ) =

P(Y = Bedroom, X= )

―――――――――――――――

P(X = )

Discriminative: Y is simple; X is always given, so not need to model P(X= )

Therefore it cannot handle missing data P(Y = Bedroom | X = )

Conditional Generative Models

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

P(X=  | Y = Bedroom)

It's often useful to condition on rich side information Y

P(X=  | Caption = "A black table with 6 chairs")

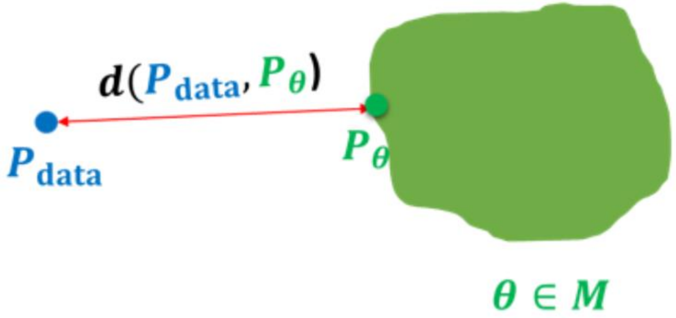
A discriminative model is a very simple conditional generative model of Y:

P(Y = Bedroom | X= )

Learning Generative Models

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

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



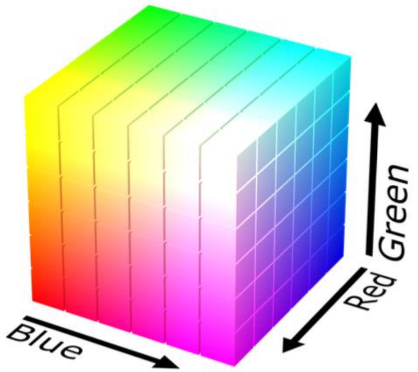
Model family

- We want to learn a probability distribution $p(x)$ over images x such that
  - **Generation:** If we sample $x_{new} \sim p(x)$, 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*)
- First question: how to represent $p(x)$

Example RGB images

Modeling a single pixel's color. Three discrete random variables:

- Red Channel R. $\text{Val}(R) = \{0, \cdots, 255\}$
- Green Channel G. $\text{Val}(G) = \{0, \cdots, 255\}$
- Blue Channel B. $\text{Val}(B) = \{0, \cdots, 255\}$



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 of Joint Distribution

- Suppose $X_1, \ldots, X_n$ are binary (Bernoulli) random variables, i.e., $\mathrm{Val}(X_i) = \{0, 1\} = \{\mathrm{Black}, \mathrm{White}\}$.
- How many possible states?

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

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

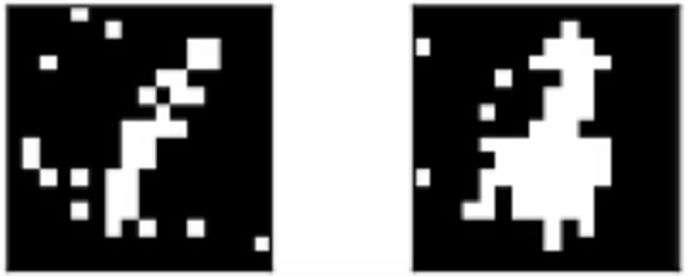
$$2^n - 1$$

Structure Through Independence

- If $X_1, \ldots, X_n$ are independent, then

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n)$$

- How many possible states? 2^n
- How many parameters to specify the joint distribution $p(x_1, \ldots, x_n)$?
  - How many to specify the marginal distribution $p(x_1)$? 1
- 2^n **entries can be described by just** n **numbers** (if $|\mathrm{Val}(X_i)| = 2$)!
- Independence assumption is too strong. Model not likely to be useful
  - For example, each pixel chosen independently when we sample from it.



Two Important Rules

1. **Chain rule** Let $S_1, \ldots S_n$ be events, $p(S_i) > 0$.

$$p(S_1 \cap S_2 \cap \cdots \cap S_n) = p(S_1)p(S_2 \mid S_1) \cdots p(S_n \mid S_1 \cap \ldots \cap S_{n-1})$$

2. **Bayes' rule** Let S_1, S_2 be events, $p(S_1) > 0$ and $p(S_2) > 0$.

$$p(S_1 \mid S_2) = \frac{p(S_1 \cap S_2)}{p(S_2)} = \frac{p(S_2 \mid S_1)p(S_1)}{p(S_2)}$$

Structure Through Conditional Independence

- Using Chain Rule

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

- 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.
  - $\cdots$
- $2^n - 1$ is still exponential, chain rule does not buy us anything.
- Now suppose $X_{i+1} \perp X_1, \dots, X_{i-1} \mid X_i$, then

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

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

Taxonomy of Generative Models

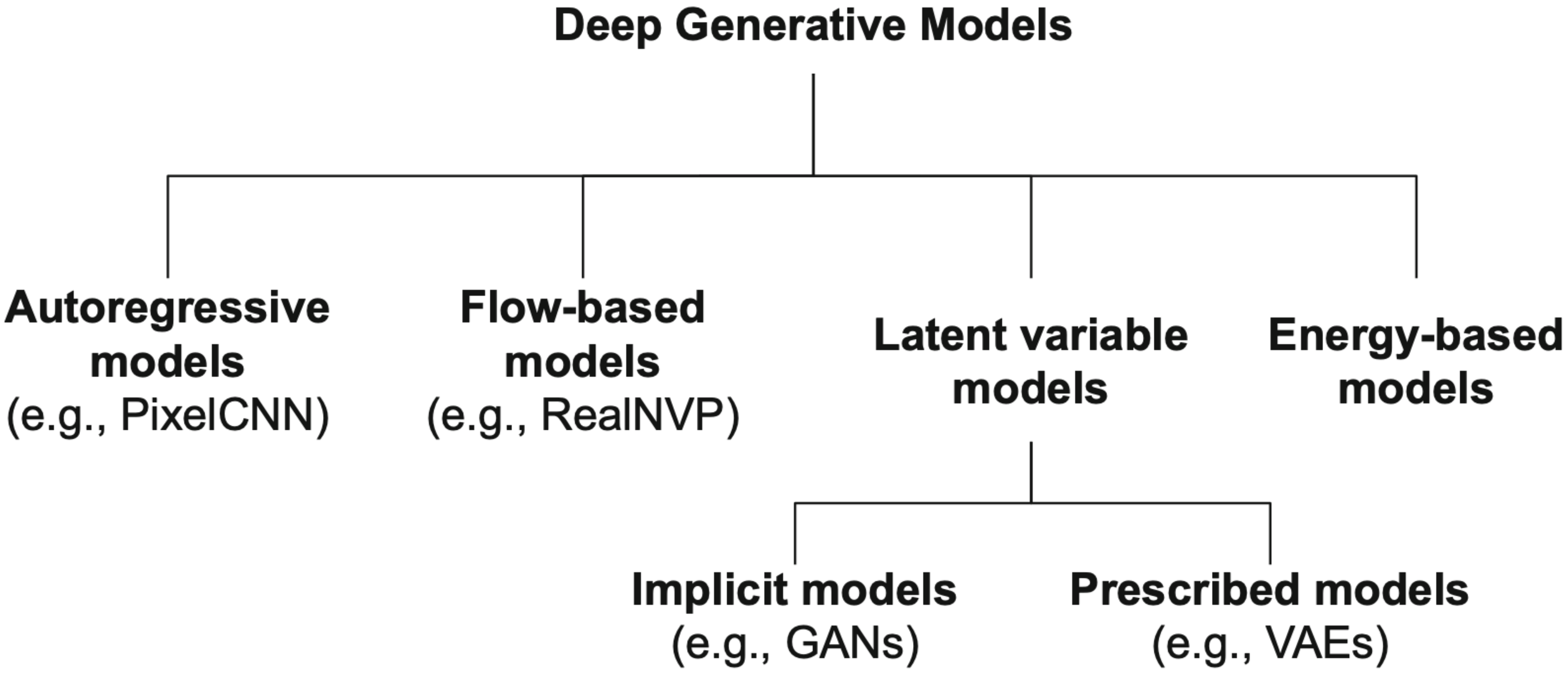


Fig. 1.4 A taxonomy of deep generative models

Taxonomy of Generative Models

- Autoregressive models

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

- Latent Variable models

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

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

- Energy Based Models

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

Latent Variable Models

- X = observed variable
- Z = latent variable

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

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

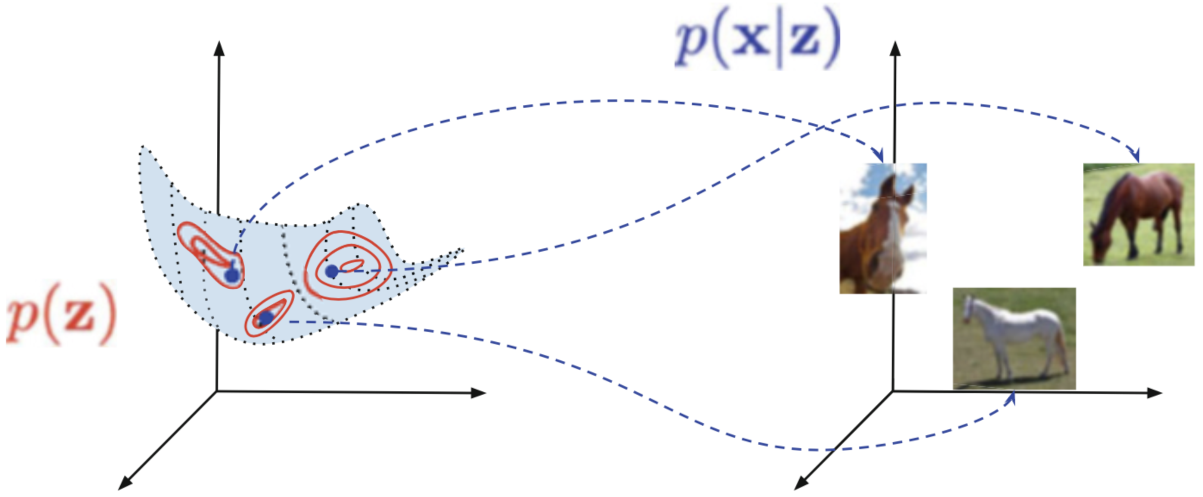


Fig. 4.1 A diagram presenting a latent variable model and a generative process. Notice the low-dimensional manifold (here 2D) embedded in the high-dimensional space (here 3D)

- Factorization of the joint model

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})$$

- Marginalization of the model

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z}$$

Maximum Likelihood Estimation (MLE)

- Suppose we have a dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, with N i.i.d. samples from the data distribution $p_\theta(\boldsymbol{x})$, parameterized by unknown θ
- i.e. $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}; \theta)$
- Our goal is to learn θ
- How to do it? Via **Maximum Log Likelihood**

Maximum Likelihood Estimation (MLE)

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

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

- Taking the log, we can rewrite

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

Maximum Likelihood Estimation (MLE)

- Hence, maximizing log likelihood is to maximizes the likelihood

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

E.g.: 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 the density of Gaussian: $p(x) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \exp(x-\mu)^\top \Sigma^{-1}(x-\mu)$
- This allows us to write down the likelihood function…

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

- … and the log of the likelihood

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

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

Finding the gradient of parameters

- Reminder: Log-likelihood objective

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

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

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

For the mean

- Reminder: Log-likelihood objective

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

- 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,

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

For the covariance

- Reminder: Log-likelihood objective

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

- 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

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

For the covariance

- Reminder: Log-likelihood objective

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

- The two facts:
  - $\boldsymbol{S} = \boldsymbol{\Sigma}^{-1}$
  - $\boldsymbol{x}^\top \boldsymbol{S}\boldsymbol{x} = \mathrm{tr}(\boldsymbol{x}^\top \boldsymbol{S}\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top)$

- Using these two facts, we can rewrite the log-likelihood in terms of $\boldsymbol{S}$ (omitting terms that derivative will cancel)

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

For the covariance

- From our re-written log-likelihood function

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

- Taking the derivative with resect 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

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

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

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

So far: Basics of Probability, Statistics

- X random variable

- X and Y independent, $p(X, Y) = p(X)\, p(Y)$

- $X_1, \ldots, X_N$ i.i.d. samples from p_θ

- Maximum likelihood estimator

- Example for Gaussian

Latent Variable Models

- X = observed variable
- Z = latent variable

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

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

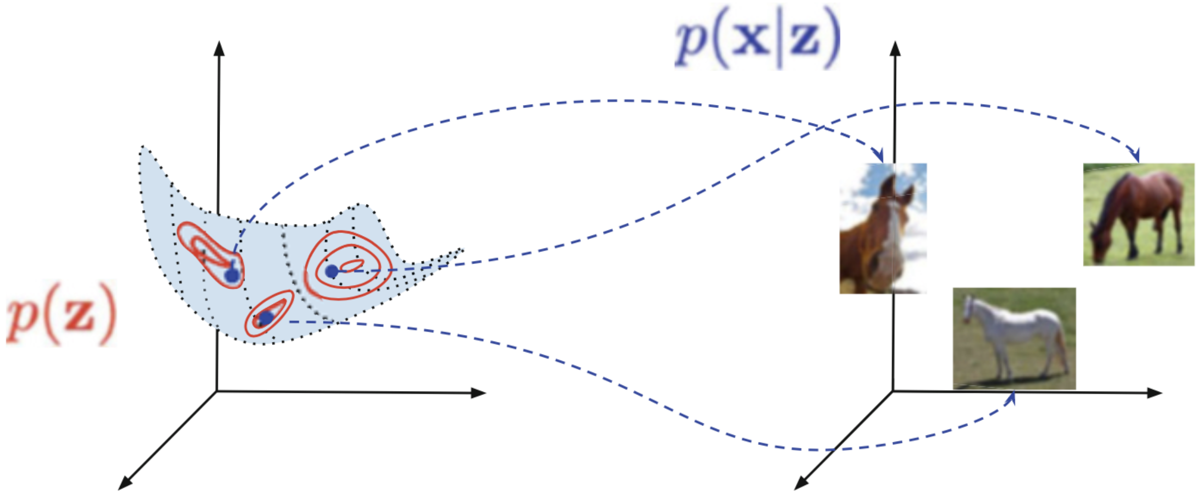


Fig. 4.1 A diagram presenting a latent variable model and a generative process. Notice the low-dimensional manifold (here 2D) embedded in the high-dimensional space (here 3D)

- Factorization of the joint model

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})$$

- Marginalization of the model

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z}$$

Latent Variable Models

- Latent Variable Model $p(x, z) = p(z)p(x \mid z)$
- To sample $p(x, z)$, we have to first
  - Sample $p(z)$
  - Then sample $p(x \mid z)$
- How to learn the parameters θ of latent variable models?
  - Let's try directly applying maximum log likelihood

$$\max_{\theta} \sum_{i=1}^{N} \log p_{\theta}(x_i) = \max_{\theta} \sum_{i=1}^{N} \log \int_{z} p_{\theta}(x_i, z) dz$$

need many samples of z for each x_i to approximate this integral when dimension is high

  - Variational Inference is our best friend here, which we will describe next

Variational Inference

- Let $q_i(z)$ be the variational distribution. Observe that
- $\sum_{i=1}^{N} \log p_\theta(x_i) = \sum_{i=1}^{N} \int q_i(z) \log p_\theta(x_i) dz = \sum_{i=1}^{N} \int q_i(z) \log \frac{p_\theta(x_i,z)}{p_\theta(z|x_i)} dz$

$$= \sum_{i=1}^{N} \int q_i(z) \log \frac{p_\theta(x_i,z)}{q_i(z)} \frac{q_i(z)}{p_\theta(z|x_i)} dz$$

$$= \underbrace{\sum_{i=1}^{N} \int q_i(z) \log \frac{p_\theta(x_i,z)}{q_i(z)} dz}_{\text{Evidence Lower Bound (ELBO)}} + \underbrace{\int q_i(z) \log \frac{q_i(z)}{p_\theta(z|x_i)} dz}_{\mathrm{KL}[q_i(z) \,||\, p_\theta(z|x_i)]}$$

$$\geq \sum_{i=1}^{N} \int q_i(z) \log \frac{p_\theta(x_i,z)}{q_i(z)} dz$$

Variational Inference

- Let $q_i(z)$ be the variational distribution. Observe that
- $\sum_{i=1}^{N} \log p_\theta(x_i) = \sum_{i=1}^{N} \int q_i(z) \log \frac{p_\theta(x_i,z)}{q_i(z)} dz + \int q_i(z) \log \frac{q_i(z)}{p_\theta(z|x_i)} dz$

  Evidence **L**ower **Bo**und (ELBO) $\mathrm{KL}[q_i(z) \,||\, p_\theta(z|x_i)]$

- Claim: $\max_{q_i : q_i(z) \geq 0, \int q_i(z) dz = 1} \int q_i(z) \log \frac{p_\theta(x_i,z)}{q_i(z)} dz = \log p_\theta(x_i)$
  - Proof: it suffices to show that $\min_{q_i : q_i(z) \geq 0, \int q_i(z) dz = 1} \mathrm{KL}[q_i(z) \,||\, p_\theta(z|x_i)] = 0$
  - Needs to dive a bit into optimization: first-order optimality conditions
- We will use VI for many latent variable models
  - Mixtures of Gaussians (a.k.a. Gaussian Mixture Models)
  - Probabilistic Principal Component Analysis (PPCA)
  - Mixtures of PPCA
  - Variational Auto-Encoders (VAE)
  - …

Expectation Maximization

$$\max_{\theta} \sum_{i=1}^{N} \log p_\theta(x_i) = \max_{\theta} \max_{w} \sum_{i=1}^{N} \int_z \; w_i(z) \log \frac{p_\theta(x_i, z)}{w_i(z)} dz$$

- Expectation Maximization alternates between two steps (k: iteration)

- E-step: $w_i^k(z) = p_{\theta_k}(z|x_i)$ maximizing w.r.t. w with θ fixed

- M-step: $\theta_{k+1} = \mathrm{argmax}_\theta \sum_{i=1}^{N} \int_z \; w_i^k(z) \log p_\theta(x_i, z)\, dz$ maximizing w.r.t. θ with w fixed

- Examples
  - For a mixture of Gaussians, E & M steps are closed-form
  - Often, E-step can be done by sampling (MCMC) and M-step can be done by optimization (SGD)

E.g.: EM for Gaussian Mixture Model

- Consider a mixture of Gaussians $p_\theta(\boldsymbol{x}) = \pi_1 p_{\theta_1}(\boldsymbol{x}) + \pi_2 p_{\theta_2}(\boldsymbol{x}) + \cdots + \pi_k p_{\theta_k}(\boldsymbol{x})$
    - $\pi_i > 0$: prior probability of drawing a point from the i-th model; $\sum_{i=1}^{k} \pi_i = 1$
    - $p_{\theta_i} = \mathcal{N}(\mu_i, \Sigma_\mathrm{i})$. $\theta_i = (\mu_i, \Sigma_\mathrm{i})$: mean and covariance of the i-th Gaussian distribution
    - $\theta = (\theta_1, \dots, \theta_k, \pi_1, \dots, \pi_k)$: the parameters of the mixture model
- Goal: estimate θ from N i.i.d. samples $x_1, \dots, x_N$ from p_θ using EM
- E-step: compute $w_{ij}^k = p_{\theta^k}\left(\boldsymbol{z}_j = i \mid \boldsymbol{x}_j\right) = \frac{p_{\theta^k}(\boldsymbol{x}_j | z_j = i) p_{\theta^k}(\boldsymbol{z}_j = i)}{p_{\theta^k}(\boldsymbol{x}_j)} = \frac{p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}{\sum_{i=1}^{n} p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}$
- M-step:
    - $\pi_i^{k+1} = \arg\max_{\pi_i} \sum_{j=1}^{N} w_{ij}^k \log(\pi_i) = \frac{\sum_{j=1}^{N} w_{ij}^k}{\sum_{j=1}^{N}\sum_{i=1}^{n} w_{ij}^k}$
    - $\theta_i^{k+1} = \arg\max_{\theta_i} \sum_{j=1}^{N} w_{ij}^k \left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^\top \Sigma_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right) - \frac{1}{2}\operatorname{logdet}(\Sigma_i)\right)$
        - $\boldsymbol{\mu}_i^{k+1} = \frac{\sum_{j=1}^{N} w_{ij}^k \boldsymbol{x}_j}{\sum_{j=1}^{N} w_{ij}^k}$ and $\Sigma_i^{k+1} = \frac{\sum_{j=1}^{N} w_{ij}^k\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1}\right)^\top}{\sum_{j=1}^{N} w_{ij}^k}$