Advanced Course on Deep Generative Models

Lecture 2: Latent Variable Models
Bayesian Inference

Dan Rosenbaum, CS Haifa

Today

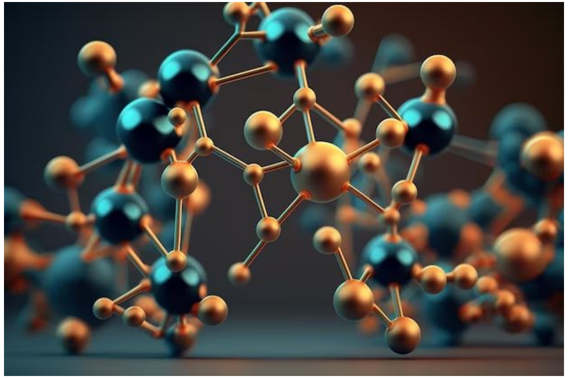
- Recap
- Introduction to Probabilistic Graphical Models
- Latent Variable Models
- Maximum likelihood
- Linear Gaussian model
- Bayesian inference

What are generative models?

- High dimensional output
- Probabilistic

ChatGPT

Generative models are a class of machine learning models designed to generate new data samples that are similar to a given dataset. These models learn the underlying structure of the data and are capable of producing new examples that mimic the characteristics of the original data.

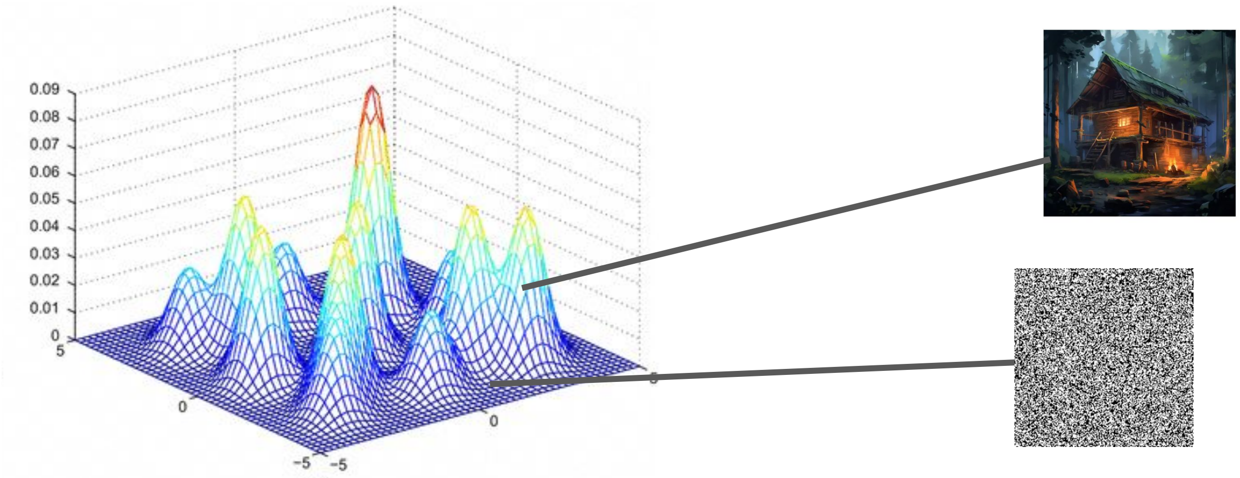


What can we do with generative models?

- Solve some task (e.g. generative classifier)
- Generate data
- Representation learning
- Measure uncertainty
- Compress
- Make decisions
- Probabilistic inference

Learning a generative model

- Data is generated by an unknown underlying distribution $\mathbf{p}_{\mathbf{data}}$
- We are looking for the parameters $\boldsymbol{\theta}$ such that $\mathbf{p}_{\boldsymbol{\theta}}$ is close to $\mathbf{p}_{\mathbf{data}}$



Components for training a generative model

1. Data - representative of the space
2. Model (e.g. Gaussian, mixture of Gaussians, Latent variable model)
3. Objective (e.g. maximum likelihood, score matching)
4. Optimization (e.g. Variational inference, MCMC)

Challenge: Solve the curse of dimensionality.

- conditional independence, structure

Probabilistic Model of an Image

- How many parameters do we need for a full parameterization of a the probability of an image?

Solutions:

- Conditional independence assumptions

- Restricted parameterizations

Discrete vs. Continuous

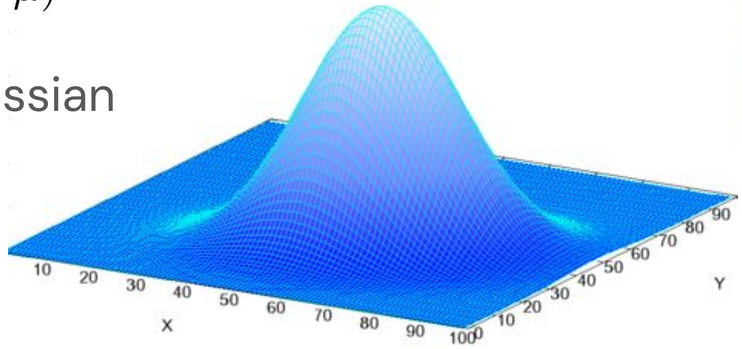
- Continuous representation is a form of assumption
  - Can lead to a more efficient representation

Multivariate Gaussians

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

Mahalanobis distance: $\Delta = (x-\mu)^T \Sigma^{-1} (x-\mu)$

1. An affine transformation of **x** is also Gaussian
2. Marginals of **x** are also Gaussian
3. Conditional distributions on some of the dimensions are also Gaussian



Gaussian models for images

Why not use a Gaussian model for images?

1. For large images we will need a huge covariance matrix

2. Distribution of images is clearly non-Gaussian

Conditional independence for images

We saw that conditional independence can reduce the number of parameters:

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

"Markov Chain" → effective for language and time series

What kind of structure should we use for images?

Conditional independence on pixels

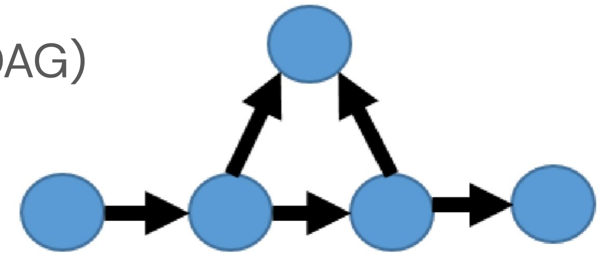
Can be effective, but not very natural structure.

Bayesian Network

- More general formulation of structure

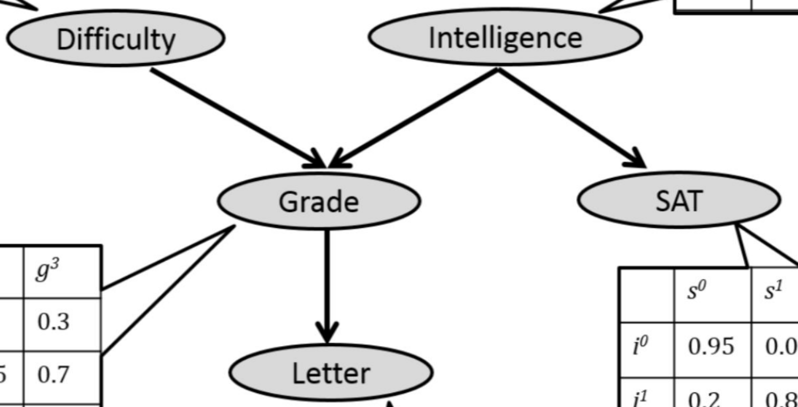
$$p(x_1, \ldots, x_n) = \prod_i p(x_i | \mathbf{x}_{\mathbf{A_i}})$$

- Can be defined with a directed acyclic graph (DAG)



- "Probabilistic Graphical Models"
- Implies a set of conditional independence assumptions

Bayesian Network Example



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Models of images

In this course

- Bayesian Networks (define structure via DAG):
  → easy to use to compute likelihoods and generate samples.
  We will use the ideas, but with very few variables (sometimes only two).

- Markov Networks (define structure via undirected graph):
  → different approach to define independence assumptions.
  → hard to transform to a valid distribution for computing likelihoods or generating samples.
  We will see this later in the course.

- Restricted parameterizations:
  → Can also be a softer way to make independence assumptions.
  We will use neural networks with different architectures.

Latent Variable Models

- Assume there’s an additional variable which we don’t see (latent, hidden)
- Use it to construct an efficient conditional independence structure.
- Simplest model for images:

- Has a more natural interpretation

- This can solve the issues we had (efficient non-Gaussian models)
  (example of models - next week)

- How do we train such models?
  (today and next week)

Training a probabilistic model

Given a model $\mathbf{p}_{\boldsymbol{\theta}}$ with unknown parameters $\boldsymbol{\theta}$, find the values of $\boldsymbol{\theta}$ that make it as close as possible to $\mathbf{p}_{\mathbf{data}}$

How can we do it?

Standard approach: Maximum likelihood.

Maximum Likelihood - Bernoulli

- X domain: **{Heads, Tails}**
- Model: **P(X = Heads) = p, P(X = Tails) = 1 – p**
- Parameter: $\boldsymbol{\theta} = \mathbf{p}$

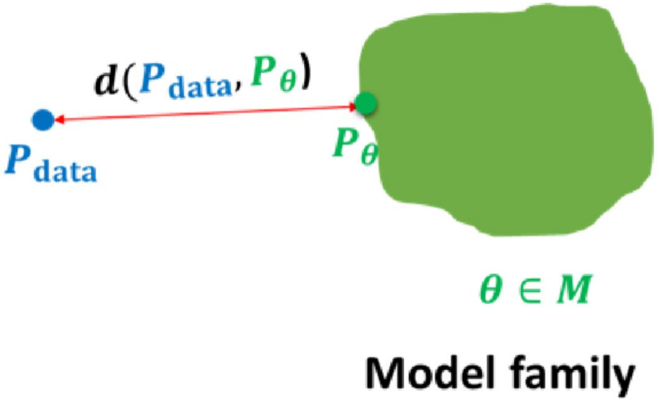
Maximum Likelihood - Gaussian

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}}$$

Justification of Maximum Likelihood

We want to minimize some distance between $\mathbf{p}_\theta$ and $\mathbf{p}_{\text{data}}$

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



How do we measure the distance between distributions?

credit: Volodymyr Kuleshov

Justification of Maximum Likelihood

KL divergence:

$$
\begin{aligned}
D_{KL}(p_{\text{data}}, p_\theta) &= \int p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_\theta(x)} dx \\
&= -\int p_{\text{data}}(x) \log p_\theta(x) dx + \text{const.} \\
&\approx -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i) + \text{const.}, \quad x_i \sim p_{\text{data}}
\end{aligned}
$$

Linear Gaussian models

Consider the problem where $\mathbf{y} = \mathbf{A}\boldsymbol{\theta} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is a random vector with a Gaussian distribution $\mathcal{N}(0,\ \mathbf{I}\,\boldsymbol{\sigma}^2)$.

Maximizing the likelihood is a general form of:

1. Linear regression with squared loss (the rows of **A** are the values of **x**)

2. Polynomial regression (the rows of **A** consist of $\mathbf{x}, \mathbf{x}^2, \mathbf{x}^2, \ldots, \mathbf{x}^k$)

Solving for $\boldsymbol{\theta}$:

Linear Gaussian models

$$
\begin{aligned}
y \sim & \mathcal{N}(A\theta, \sigma^2 I) \\
\frac{\partial \log p_\theta(y)}{\partial \theta} =& \frac{1}{\sigma^2} A^\top (y - A\theta) \\
\frac{\partial \log p_\theta(y)}{\partial \theta} =& 0 \\
A^\top =& A^\top A\theta \\
\hat{\theta}_{\mathrm{M}L} =& (A^\top A)^{-1} A^\top y
\end{aligned}
$$

Latent Variables and Bayesian Statistics

- So far we've used the classical (frequentist) approach:
  assume there is a single true value for the parameters $\boldsymbol{\theta}$

- Using latent variables makes the estimation closer to Bayesian statistics

- We will first understand Bayesian statistics, and then look at the connection to latent variable models.

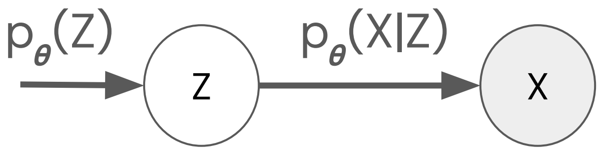
The Bayesian Philosophy

- Every unknown value is a random variable
- We always maintain a distribution over random variable ⇒ “belief”
- Given some observation, the distribution is updated via Bayes’ rule

Advantages: capture uncertainty, can define optimal estimators.

Disadvantages: assumes a prior, can be hard to compute.

Back to Latent Variable Models



- We want models with latent variables to make them efficient and powerful

- We have an unknown latent variable + unknown parameters.

- How can we train them?

  Semi-Bayesian approach:
  - Unknown latent variable **Z** + unknown parameters $\boldsymbol{\theta}$.
  - Maximum likelihood for $\boldsymbol{\theta}$, Bayesian for **Z**