Advanced Course on Deep Generative Models

Dan Rosenbaum

Course Structure

1. Advanced Course
2. First part:

   Theoretical + Classical generative models
3. Second part:

   Deep generative models - concepts + implementations

Prior Knowledge

You should be familiar with:

1. Probability
2. Linear algebra
3. Calculus
4. Machine learning
5. Deep learning

Resources

- Course on Probabilistic Modeling and Reasoning (University of Edinburgh)
  https://www.inf.ed.ac.uk/teaching/courses/pmr/22-23/

- Course on Deep Generative Models (Stanford)
  https://www.youtube.com/watch?v=XZOPMRWXBEU

- Last chapters in the Deep Learning Book by Goodfellow, Bengio & Courville
  https://www.deeplearningbook.org/

Grade

1. Home assignments (submission in pairs):
    - 3 theoretical assignments
    - 4 coding assignments
2. Final Quiz
3. Attendance

Today

- Introduction to deep generative models
- Basic concepts in learning generative models (probability)
- Multivariate Gaussian distribution

Deep Generative Models

What are generative models?

- High dimensional output
- Probabilistic

ChatGPT

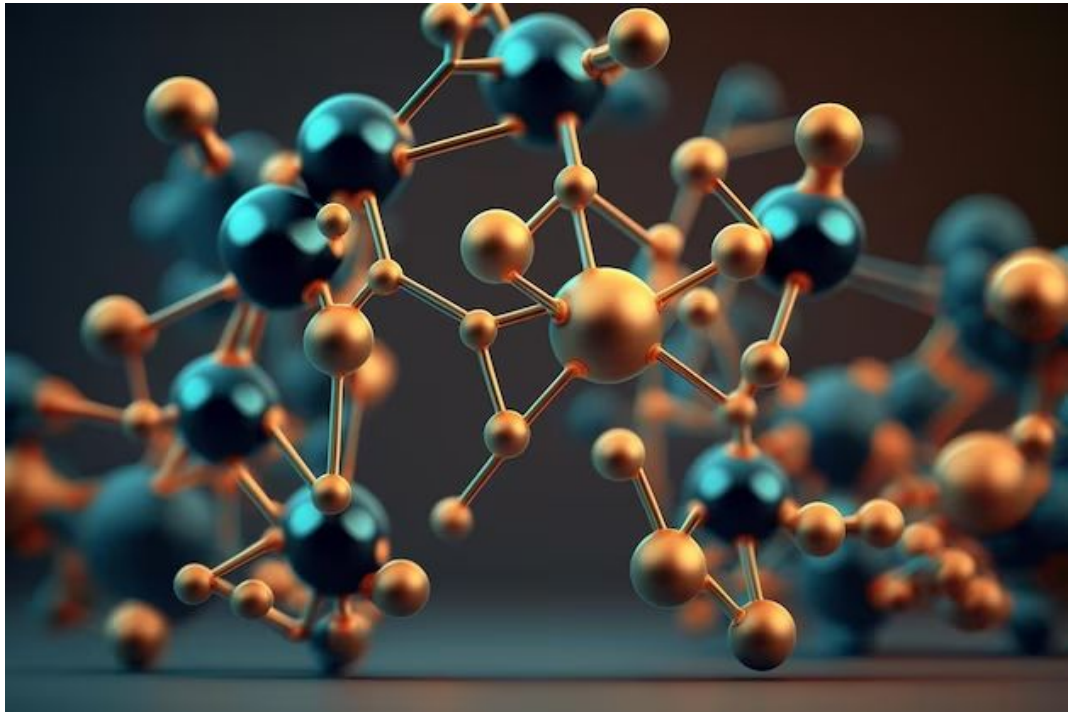
Generative models are a class of machine learning models designed to generate new data samples that are similar to a given dataset. These models learn the underlying structure of the data and are capable of producing new examples that mimic the characteristics of the original data.

Audio:

Video:

https://openai.com/sora/#features

Chemical molecules



Classical Supervised Learning

Discriminative

- Directly train a model of Y | X

P(Y | X)

- Y is usually low dimensional

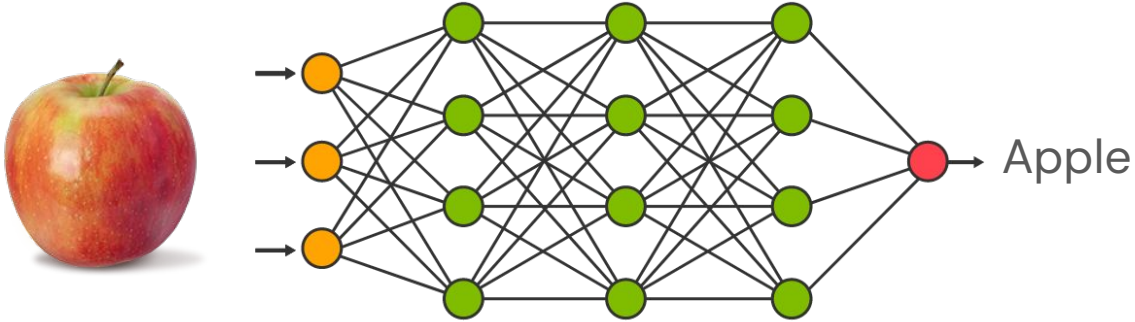
Generative

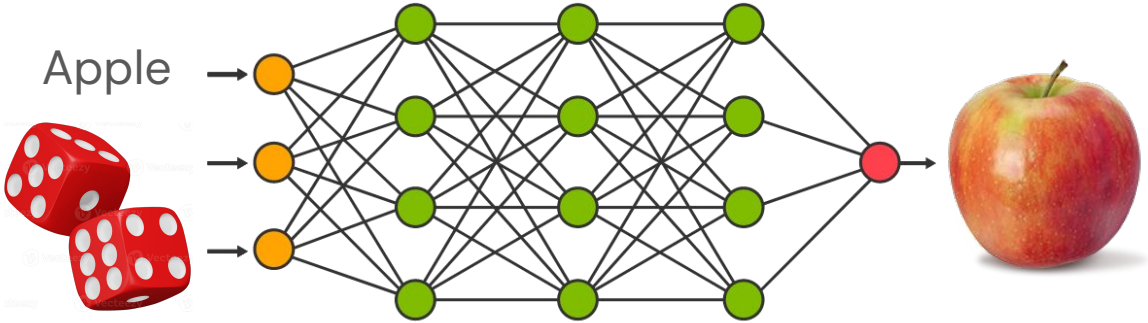
- Train a model of X | Y

P(X | Y)

- X is usually high dimensional
- Predict Y using Bayes' rule

Discriminative vs. Generative models





Discriminative vs. Generative models

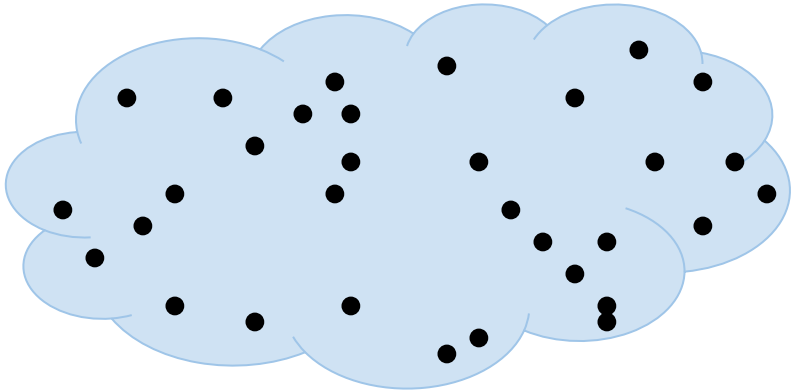
- Generative models can do what discriminative models do – use Bayes' rule to compute **P(Y|X)** from **P(X|Y)**

- But generative models can do much more:
  - Generate new data **x**
  - Generate conditional data **x|y**
  - Handle missing information in **x**
  - And much more.

Unsupervised Learning

- Train an unconditional model that captures the distribution of data

P(X)



- Access to much more data

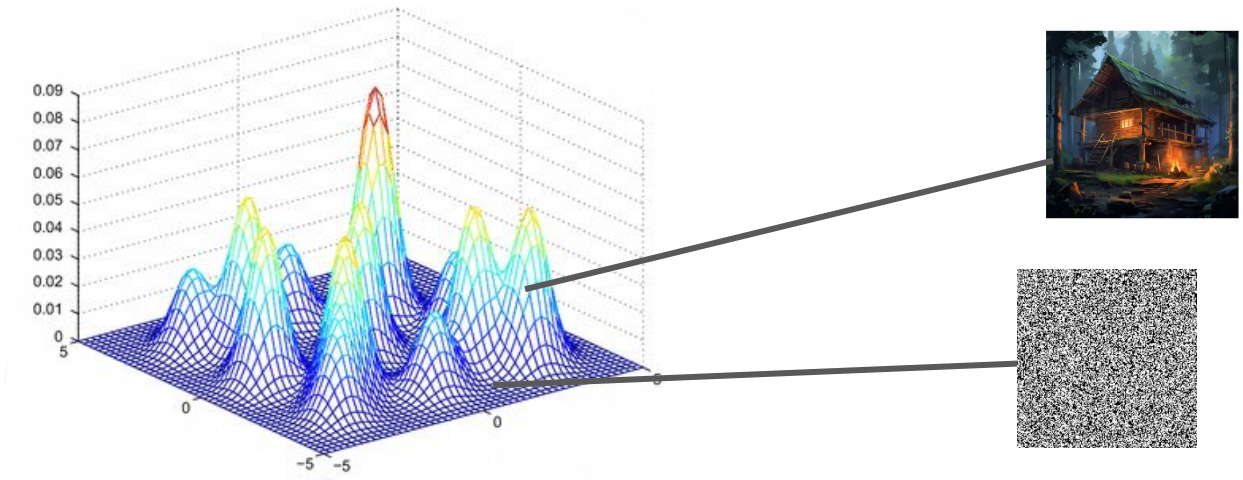
Hand-Crafted vs. Learned

- We can come up with a set of rules to characterize or generate data
  - Smoothness
  - Graphics engine

- Here we are interested in models that are learned from data

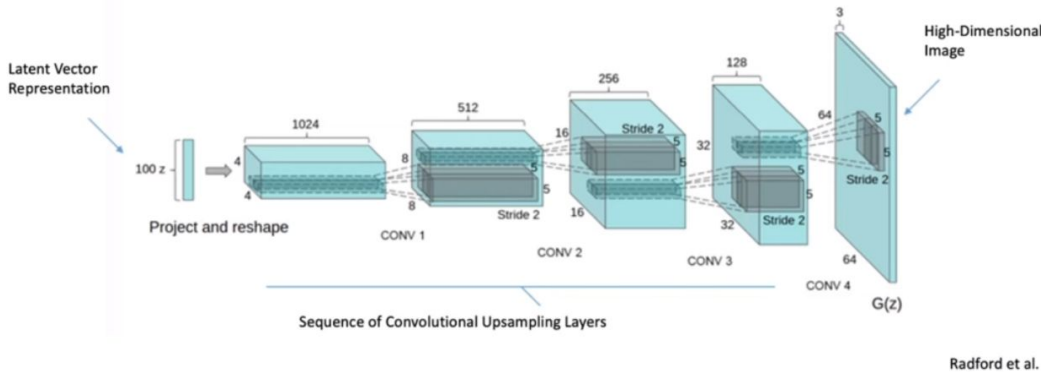
Probability

- The language we use to specify desired vs. undesired data points
- Soft assignment
- The goal of training: capture the distribution of the training data



Deep Generative Models

- The distribution of high dimensional data (e.g. images) can be very complex
- We will use deep neural networks to model the distribution



Radford et al.

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)

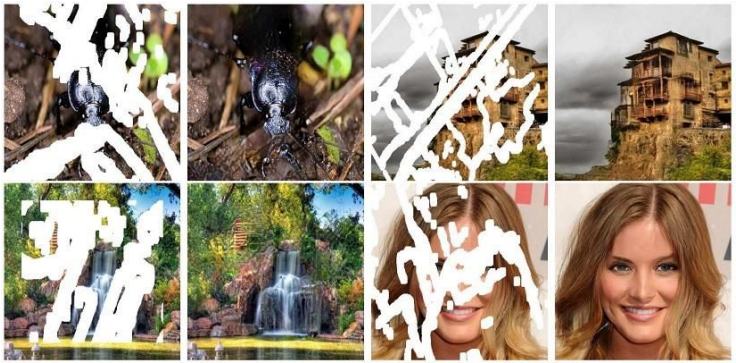
What can we do with generative models?

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

Art

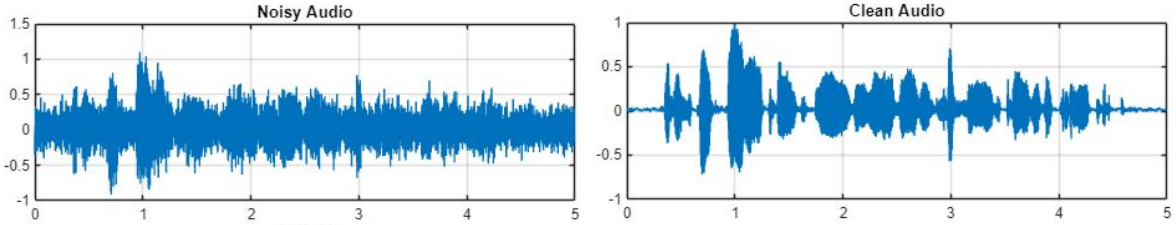
Signal processing

Inpainting

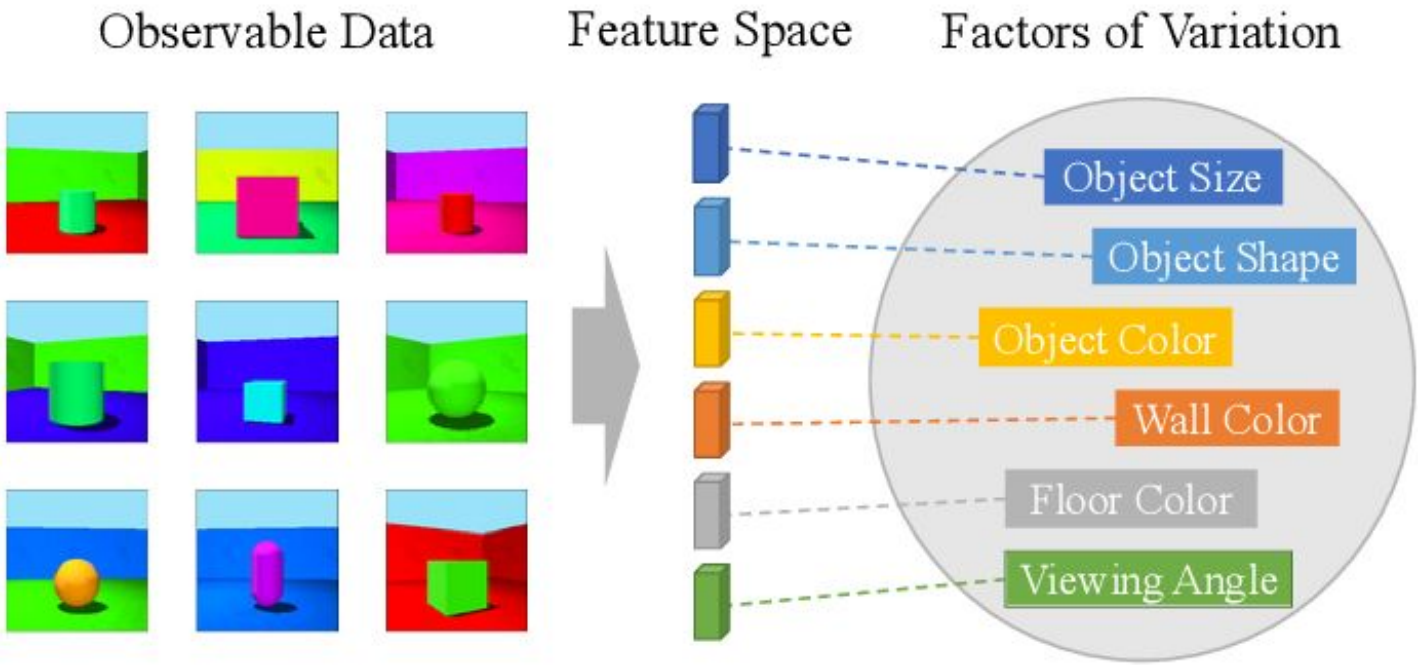


Super resolution

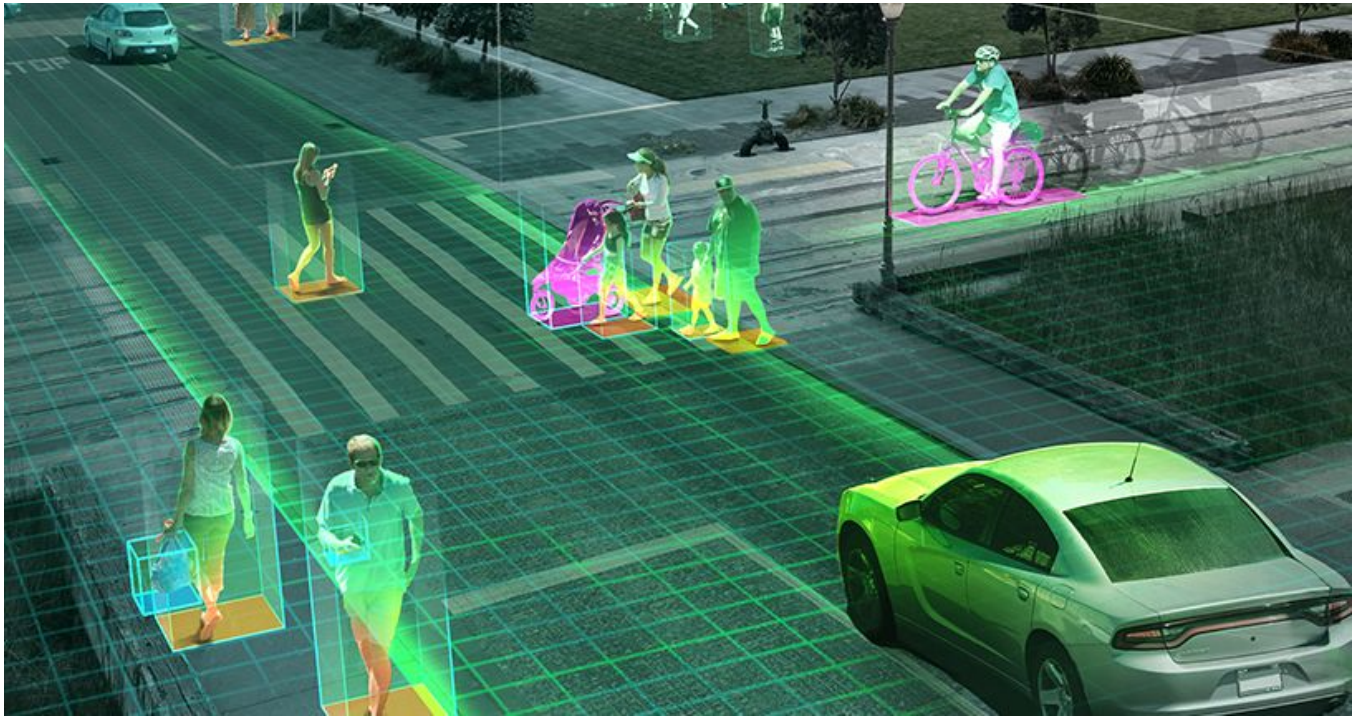
Audio denoising



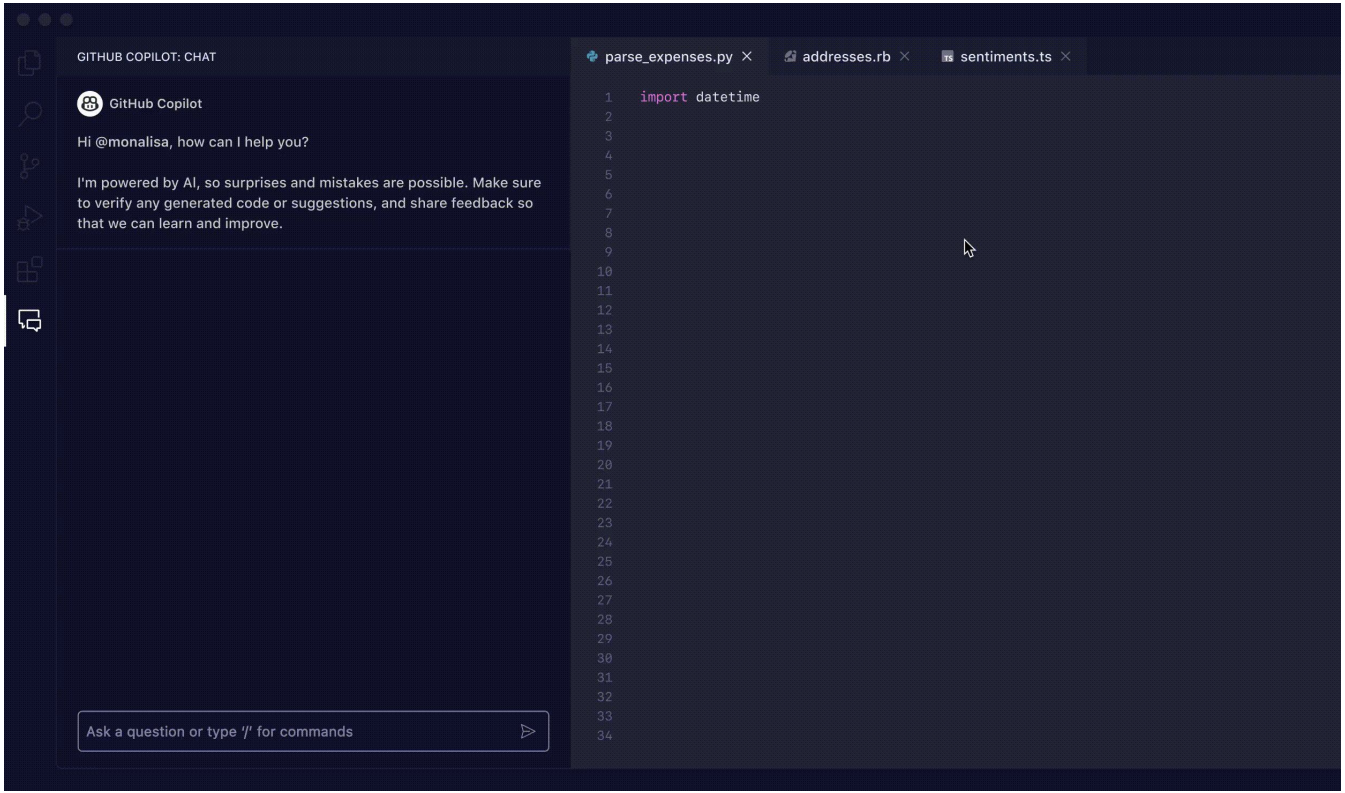
Representation learning



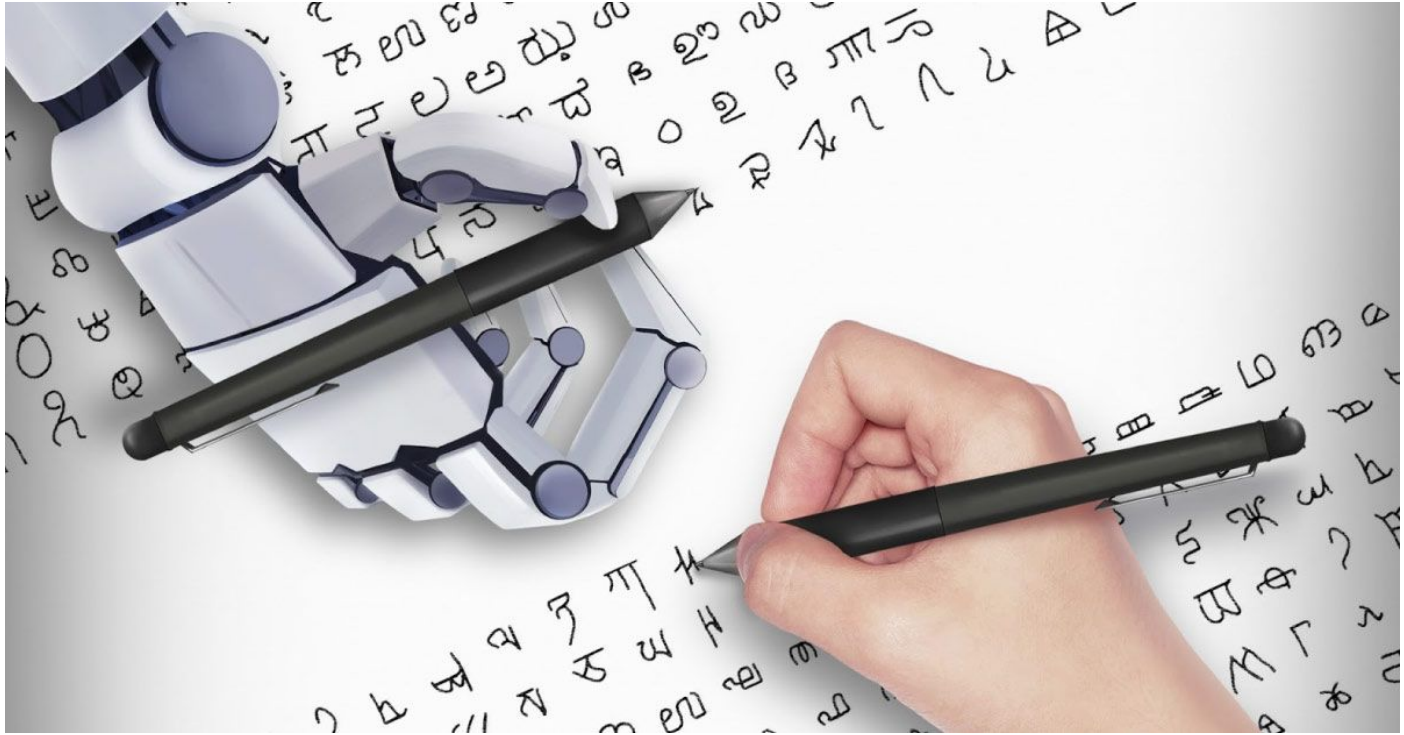
Decision making



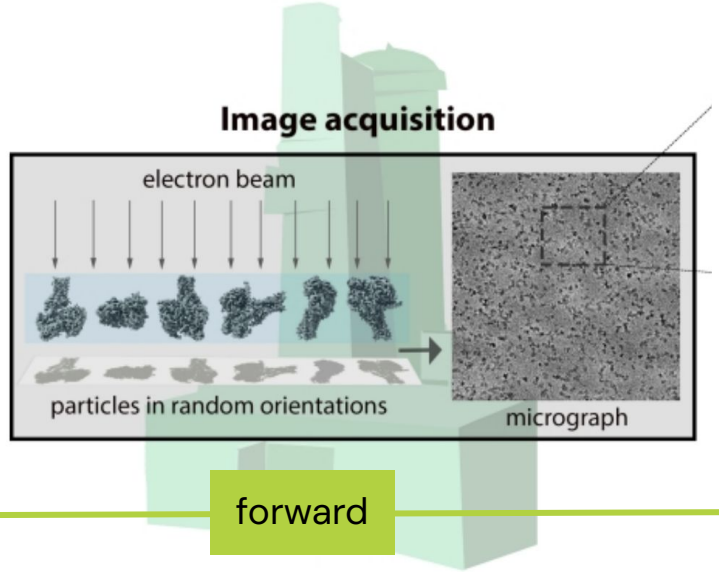
Code assistant

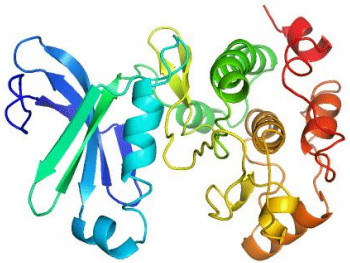


Translation

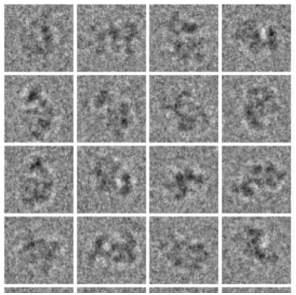


Science





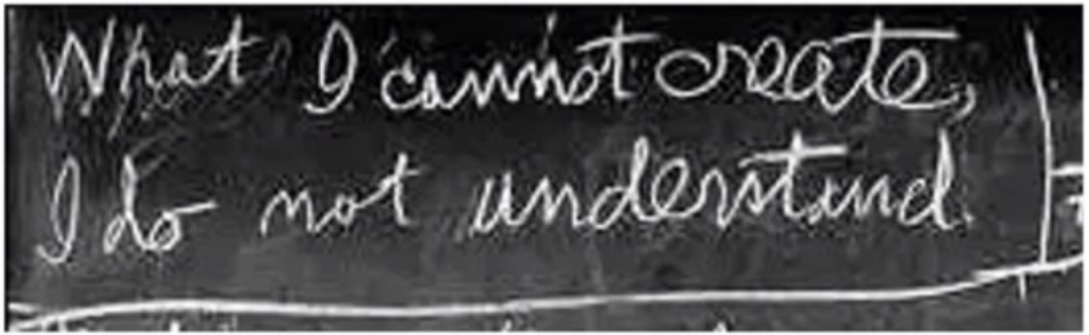
forward

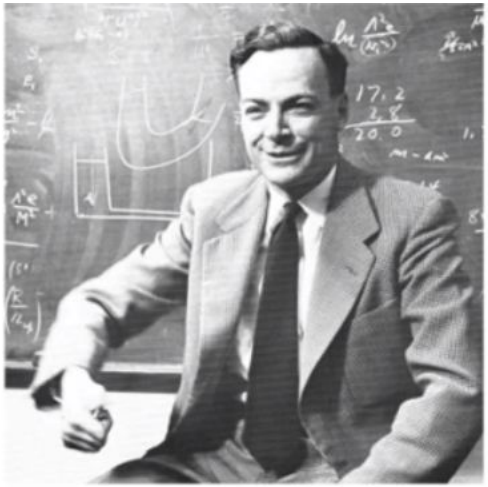


inverse

Inferring a Continuous Distribution of Atom Coordinates from Cryo-EM Images using VAEs, Rosenbaum et al. 2021

Generative models as a path to AI





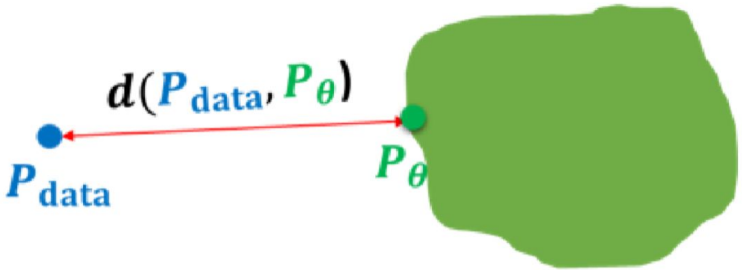
Richard Feynman: “*What I cannot create, I do not understand*”

Learning a Generative Model

Machine Learning Setup

- 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 as close as possible to $\mathbf{p}_{\mathbf{data}}$

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

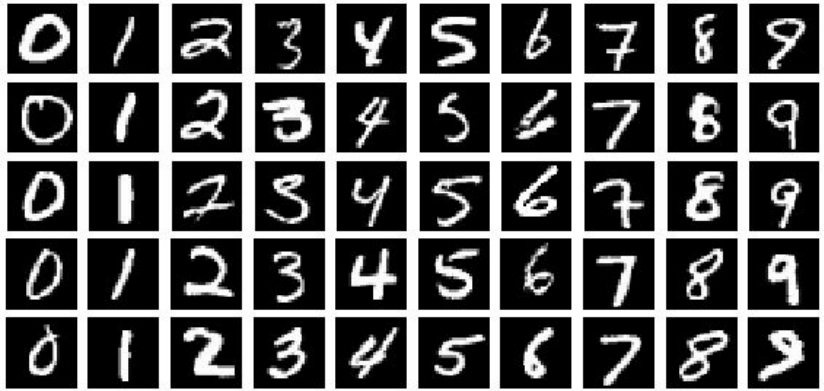


Modeling p_θ

Bernoulli distribution: biased coin flip

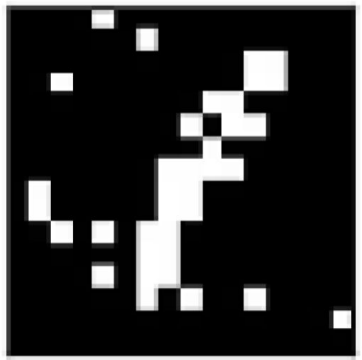
- X domain: **{Heads, Tails}**
- Model: **P(X = Heads) = p, P(X = Tails) = 1 – p**
- Parameter: θ **= p**

- Categorical distribution?
- Multivariate categorical distribution?
- How many parameters?
- Curse of dimensionality



Independence

- If the variables $x_1, \ldots, x_n$ are independent.
- How many possible states? How many parameters?
- Independence assumption is too strong



Conditional Independence

- Two events A, B are conditionally independent given event C if

$$p(A \cap B|C) = p(A|C)p(B|C)$$

- Random variables X, Y are conditionally independent given Z if for all values $x \in \text{Val}(X)$, $y \in \text{Val}(Y)$, $z \in \text{Val}(Z)$

$$p(X = x \cap Y = y|Z = z) = p(X = x|Z = z)p(Y = y|Z = z)$$

- We will also write $p(X, Y|Z) = p(X|Z)p(Y|Z)$. Note the more compact notation.
- Equivalent definition: $p(X|Y, Z) = p(X|Z)$.
- We write $X \perp Y \mid Z$

Two important rules in probability

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

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

- 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.
  - $\cdots$
- $2^n - 1$ is still exponential, chain rule does not buy us anything.

Structure through conditional independence

- Now suppose $X_{i+1} \perp X_1, \ldots, X_{i-1} \mid X_i$, then

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

- How many parameters?

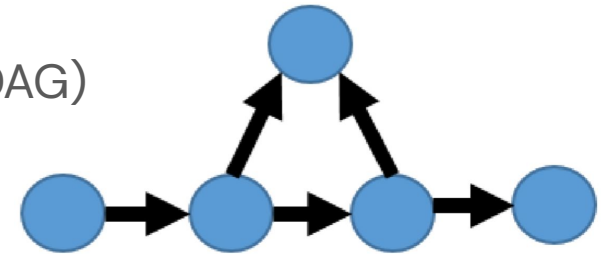
$2n - 1$. Exponential reduction!

Structure through conditional independence

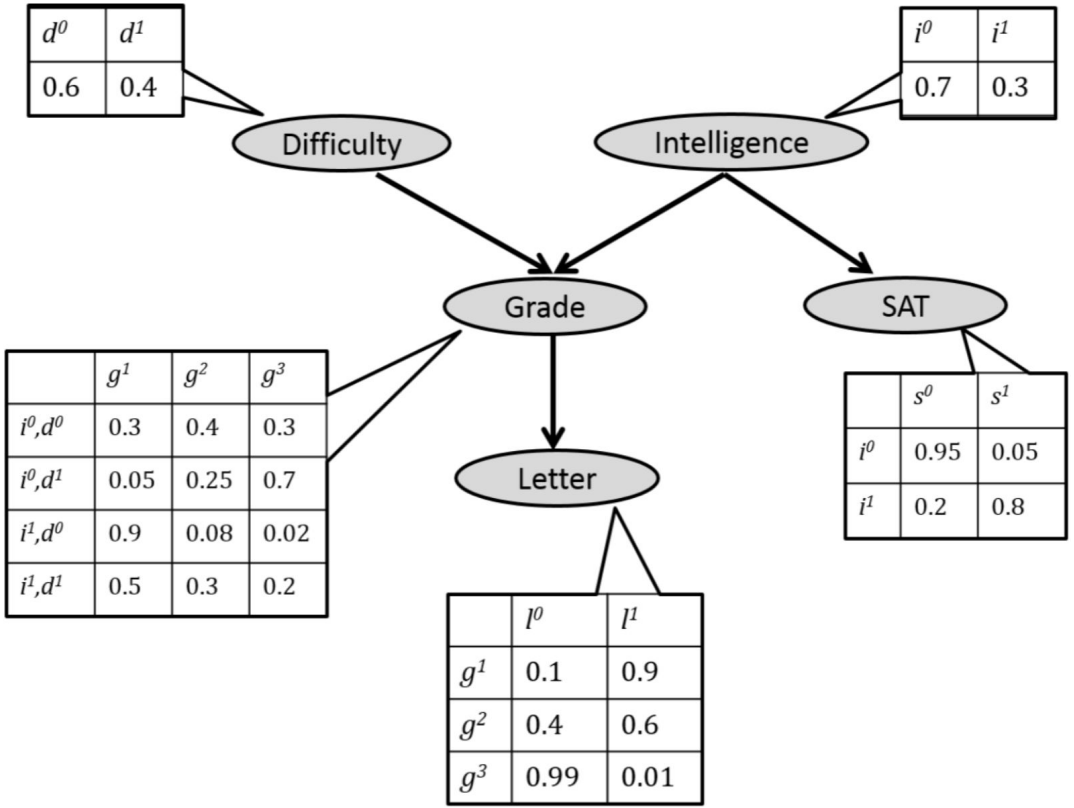
- More general formulation - Bayesian network:

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

- Has to correspond to a chain-rule factorization.
- Can be defined with a directed acyclic graph (DAG)
- “Probabilistic Graphical Models”



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

Useful formula

- **X** is a continuous random variable with pdf $\mathbf{p_x(x)}$,
- **Y** is defined as **f(X)**
- What is the pdf of **Y**?

If **f** is monotonous and invertible:

$$p_y(y) = \left| \frac{\partial f^{-1}(y)}{\partial y} \right| p_x\left(f^{-1}(y)\right)$$

change of variable formula

Proof

For the case when **f** is monotonically increasing:

$$P_y(y) \triangleq P(Y \le y) = \int_{-\infty}^{y} p_y(\tilde{y})\, d\tilde{y}$$

$$P(Y \le y) = P(f(X) \le y) = P(X \in \{x \mid f(x) \le y\})$$
$$= P\left(X \le f^{-1}(y)\right) = P_x\left(f^{-1}(y)\right)$$

differentiating:

$$p_y(y) \triangleq \frac{\partial}{\partial y} P_y(y) = \frac{\partial}{\partial y} P_x\left(f^{-1}(y)\right) = \frac{\partial f^{-1}(y)}{\partial y} \frac{\partial}{\partial f^{-1}(y)} P_x\left(f^{-1}(y)\right)$$
$$= \frac{\partial f^{-1}(y)}{\partial y} p_x\left(f^{-1}(y)\right)$$

Example

Let **z = x²** where the PDF of **x** is:

$$p_x(x) = \begin{cases} \frac{1}{\alpha} & x \in [0,\ \alpha] \\ 0 & \text{otherwise} \end{cases}$$

What is the PDF of z?

In the range **[0, *α*] f** is invertible: $x = f^{-1}(z) = \sqrt{z}$

The derivative of the inverse is: $\frac{\partial f^{-1}(z)}{\partial z} = \frac{1}{2} \cdot \frac{1}{\sqrt{z}}$

$$\Rightarrow \quad p_z(z) = \frac{1}{2} \cdot \frac{1}{\sqrt{z}} p_x\left(\sqrt{z}\right) = \begin{cases} \frac{1}{2\alpha} \cdot \frac{1}{\sqrt{z}} & 0 \le \sqrt{z} \le \alpha \\ 0 & \text{otherwise} \end{cases}$$

Multivariate distributions

We can define a distribution $\mathbf{x \sim p(x)}$ where **x** is a multi-dimensional vector

$$\mathbf{x = [x_1, \ldots, x_n]}$$

$\Rightarrow$ The expectation of **x** is a vector $\mathbf{[\ E[x_1], \ldots, E[x_n]\ ]}$

$\Rightarrow$ The covariance of **x** is a matrix $\mathbf{\Sigma}$ where $\mathbf{\Sigma_{ij} = cov[x_i\, x_j]}$

- $\mathbf{\Sigma}$ is symmetric, PD, invertible.

Multivariate change of variable

$$p_y(\boldsymbol{y}) = p_x\left(f^{-1}(\boldsymbol{y})\right)\left|J_y\left[f^{-1}(\boldsymbol{y})\right]\right|$$

Jacobian: a matrix defining all the partial derivatives of a multivariate function

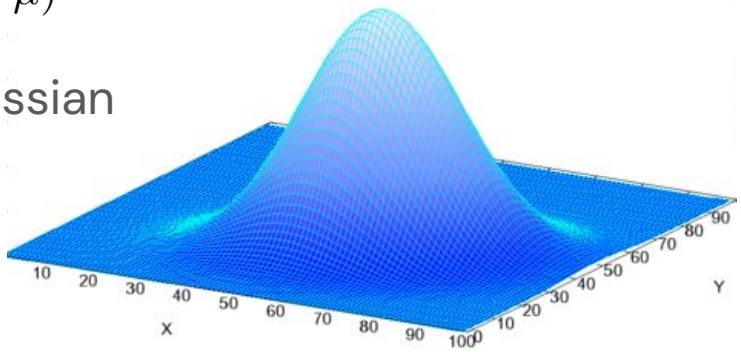
$$\left[J_x\left[f(x)\right]\right]_{ij} \triangleq \frac{\partial\left[f(x)\right]_i}{\partial x_j}$$

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



Decomposing the covariance

- Decomposing x into two parts: $x = \begin{pmatrix} x_a \\ x_b \end{pmatrix} \Rightarrow \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$

- Defining the precision matrix: $\Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$

- Using the identity: $\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} M & -MBD^{-1} \\ -D^{-1}CM & D^{-1} + D^{-1}CMBD^{-1} \end{pmatrix}$

  Where **M = (A–BD^{-1}C)$^{-1}$**

  "Schur's complement"

$$\Rightarrow \quad \begin{aligned} \Lambda_{aa} &= \left(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}\right)^{-1} \\ \Lambda_{ab} &= -\Lambda_{aa}\Sigma_{ab}\Sigma_{bb}^{-1} \\ \Lambda_{bb} &= \left(\Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}\right)^{-1} \end{aligned}$$

Completing the Square

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

Completing the Squares

Claim: if $p(x) \propto \exp\left(-\frac{1}{2}(x^T A x + 2x^T b + c)\right)$ (general quadratic)

$\Rightarrow p(x)$ is Gaussian

proof: (for the case of invertible A)

We want to show

$$p(x) \propto \exp\left(-\frac{1}{2}(x-\mu)^T \Lambda (x-\mu)\right)$$

$$p(x) \propto \exp\left(-\tfrac{1}{2}(x-\mu)^T \Lambda (x-\mu)\right)$$

$$x^T A x + 2x^T b + c = x^T A x + 2x^T A A^{-1} b + c$$

$$= x^T A x + 2x^T A A^{-1} b + b^T A^{-1} A A^{-1} b - \underbrace{b^T A^{-1} A A^{-1} b + c}_{\text{does not depend on } x}$$

$$= (x - A^{-1}b)^T A (x - A^{-1}b) + \tilde{c}$$

$$\Rightarrow \Lambda = A, \quad \Sigma = A^{-1}, \quad \mu = A^{-1} b$$

proofs: ② ③

Given: $\begin{pmatrix} x_a \\ x_b \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \right)$

show that:

③ $p(x_b|x_a)$ is Gaussian

② $p(x_a)$ is Gaussian

The plan:

- compute $p(x_b|x_a)$ by treating x_a as const.
- compute $p(x_a) = f(x_a) \int p(x_b|x_a) dx_b$

Simplify: remove $\mu_a, \mu_b \Rightarrow y = x - \mu$

$$\Delta = y^T \Lambda y = \begin{pmatrix} y_a \\ y_b \end{pmatrix}^T \begin{pmatrix} \Lambda_a & B \\ B^T & \Lambda_b \end{pmatrix} \begin{pmatrix} y_a \\ y_b \end{pmatrix}$$

$$= \begin{pmatrix} y_a \\ y_b \end{pmatrix}^T \begin{pmatrix} \Lambda_a y_a + B y_b \\ B^T y_a + \Lambda_b y_b \end{pmatrix}$$

$$\doteq y_a^T \Lambda_a y_a + \underbrace{2 y_b^T B^T y_a + y_b^T \Lambda_b y_b}_{*}$$

quadratic $\Rightarrow$ Gaussian

$$* = y_b^T \Lambda_b y_b + 2 y_b^T \Lambda_b \Lambda_b^{-1} B^T y_a + y_a^T B \Lambda_b^{-1} \Lambda_b \Lambda_b^{-1} B^T y_a - y_a^T B \Lambda_b^{-1} \Lambda_b \Lambda_b^{-1} B^T y_a$$

$$(y_b - \Lambda_b^{-1} B^T y_a)^T \Lambda_b (y_b - \Lambda_b^{-1} B^T y_a)$$

$$* = y_b^T \Lambda_b y_b + 2 y_b^T \Lambda_b \Lambda_b^{-1} B^T y_a + y_a^T B \Lambda_b^{-1} \Lambda_b \Lambda_b^{-1} B^T y_a - y_a^T B \Lambda_b^{-1} \Lambda_b \Lambda_b^{-1} B^T y_a$$

$$(y_b - \Lambda_b^{-1} B^T y_a)^T \Lambda_b (y_b - \Lambda_b^{-1} B^T y_a)$$

$$\Rightarrow E[y_b | y_a] = \Lambda_b^{-1} B^T y_a \qquad E[X_b | X_a] = \mu_b + \Lambda_b^{-1} B^T (X_a - \mu_a)$$

$$Cov[y_b | y_a] = Cov[X_b | X_a] = \Lambda_b^{-1} = \Sigma_{bb} - \Sigma_{ba} \Sigma_{aa}^{-1} \Sigma_{ab}$$

$$E[X_b | X_a] = \mu_b + \Sigma_{ba} \Sigma_{aa}^{-1} (X_a - \mu_a)$$

■ (3)

To prove (2) take all terms in Δ that depend on y_a and is not in $p(y_b)$.

$$f(y_a) = y_a^T \Lambda_a y_a - y_a B \Lambda_b^{-1} B^T y_a$$

$$= y_a^T \left(\Lambda_a - B\Lambda_b^{-1} B^T\right) y_a$$

quadratic $\Rightarrow$ Gaussian

$$E[x_a] = \mu_a$$

$$\mathrm{Cov}[y_a] = \left(\Lambda_a - B\Lambda_a B^T\right)^{-1} = \Sigma_{aa}$$

■ (2)