Advanced Topics in Machine Learning (deep generative modelling)

Lecture 2: Distribution approximation

Nikolay Malkin

20 January 2026

Outline of Lecture 2

Maths review + generative modelling as optimisation:

- Some notes and review of Lecture 1
- Preliminaries
  - Probability distributions and density functions
  - Generative processes
- Generative modelling as an optimisation problem
  - Divergence measures

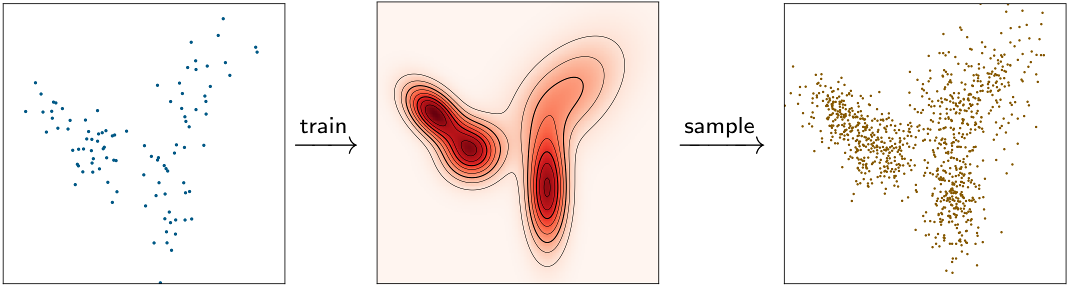
- Some notes and review of Lecture 1
- Preliminaries
  - Probability distributions and density functions
  - Generative processes
- Generative modelling as an optimisation problem
  - Divergence measures

Admin notes

- Sample exam available on website
  - It is 45 marks, but the real exam will be 25
- Slides published the evening before each lecture
- Tutorial for this track: Mondays 13:10-14:00 (NM present) and 14:10–15:00 (KT present), Appleton Tower Teaching Studio M2
- Tutorial materials published at end of preceding week
  - Sooner in future weeks
  - Theory and programming parts; come prepared with questions!

Lecture 1 review

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Today, we make this formal.)



Lecture 1 review

- A **generative model** is a probability distribution, or generative process, that is **derived from data** so as to approximate the distribution that produced the data.
- A **deep generative model** is one that uses **deep neural networks** to represent (components of) the generative process.
- A **deep generative modelling algorithm** consists of: a choice of generative process, a family of distributions parametrised by neural networks to represent that process, and a **learning algorithm** to fit those networks' parameters to data.

Lecture 1 review

- A **generative model** is a probability distribution, or generative process, that is **derived from data** so as to approximate the distribution that produced the data.
- A **deep generative model** is one that uses **deep neural networks** to represent (components of) the generative process.
- A **deep generative modelling algorithm** consists of: a choice of generative process, a family of distributions parametrised by neural networks to represent that process, and a **learning algorithm** to fit those networks' parameters to data.

The questions:

- How to represent the approximating distribution (i.e., the choice of generative process and its parametrisation)
- How to fit it to data (the learning algorithm)

Lecture 1 review

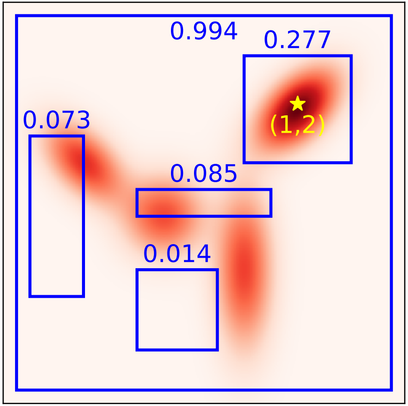
Desiderata for generative modelling:

- Fidelity (samples should look like training data)
  - The model should not produce samples far from the training data with high probability
- Diversity (samples should represent the variation in the training data)
  - The model should produce samples close to all parts of the training data with high probability
- Novelty (samples should not be copies of training data)
  - The modelled distribution should be smooth to prevent memorisation (overfitting)

- Some notes and review of Lecture 1
- Preliminaries
  - Probability distributions and density functions
  - Generative processes
- Generative modelling as an optimisation problem
  - Divergence measures

Probability distributions

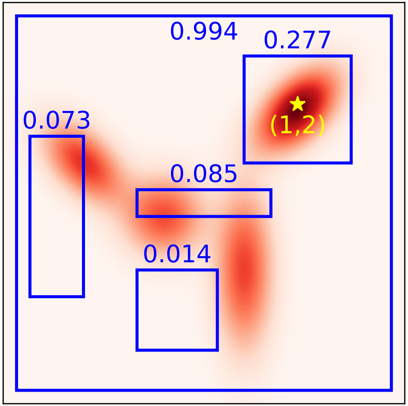
- A **probability distribution** μ over $\mathbb{R}^d$ is a function that assigns a number $\mu(A) \geq 0$ to every measurable subset A of $\mathbb{R}^d$, satisfying certain axioms
  - Such subsets A are called **events**
  - Axiom: $\mu(\mathbb{R}^d) = 1$, $\mu(\emptyset) = 0$
  - Axiom: if $A_1 \cap A_2 = \emptyset$, then $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2)$
  - (We do not discuss the details here; measure theory studies this in depth.)
- Meaning: $\mu(A)$ is the probability that a random sample $X \sim \mu$ lies in A



Probability distributions

- A **probability distribution** μ over $\mathbb{R}^d$ is a function that assigns a number $\mu(A) \geq 0$ to every measurable subset A of $\mathbb{R}^d$, satisfying certain axioms
- Meaning: $\mu(A)$ is the probability that a random sample $X \sim \mu$ lies in A
- Some probability distributions can be described by **density functions** $p : \mathbb{R}^d \to [0, \infty)$; in this case:

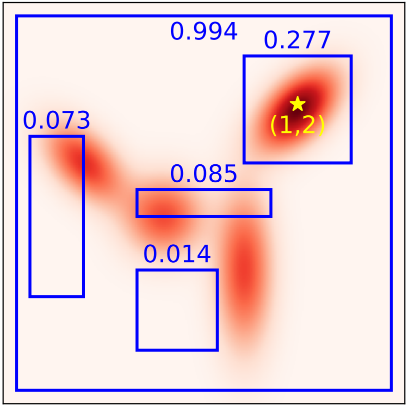
$$\mu(A) = \int_A p(x)\, \mathrm{d}x = \int_{\mathbb{R}^d} \mathbf{1}[x \in A] p(x)\, \mathrm{d}x$$



Probability distributions

- A **probability distribution** μ over $\mathbb{R}^d$ is a function that assigns a number $\mu(A) \geq 0$ to every measurable subset A of $\mathbb{R}^d$, satisfying certain axioms
- Meaning: $\mu(A)$ is the probability that a random sample $X \sim \mu$ lies in A
- Some probability distributions can be described by **density functions** $p : \mathbb{R}^d \to [0, \infty)$; in this case:

$$\mu(A) = \int_A p(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} \mathbf{1}[x \in A]p(x)\,\mathrm{d}x$$



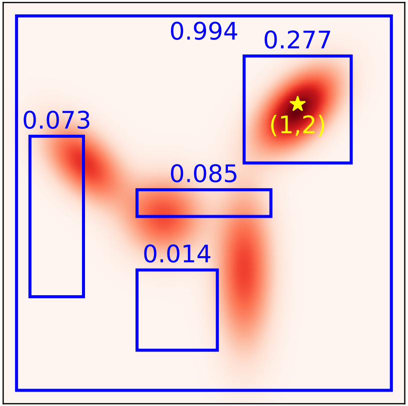
What is $p(\{(1,2)\})$?

Probability distributions

- A **probability distribution** μ over $\mathbb{R}^d$ is a function that assigns a number $\mu(A) \geq 0$ to every measurable subset A of $\mathbb{R}^d$, satisfying certain axioms
- Meaning: $\mu(A)$ is the probability that a random sample $X \sim \mu$ lies in A
- Some probability distributions can be described by **density functions** $p : \mathbb{R}^d \to [0, \infty)$; in this case:

$$\mu(A) = \int_A p(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} \mathbf{1}[x \in A] p(x)\,\mathrm{d}x$$

What is $p(\{(1,2)\})$? 0 (points have zero probability mass under continuous distributions).



Probability distributions

- A **probability distribution** μ over $\mathbb{R}^d$ is a function that assigns a number $\mu(A) \geq 0$ to every measurable subset A of $\mathbb{R}^d$, satisfying certain axioms
- Meaning: $\mu(A)$ is the probability that a random sample $X \sim \mu$ lies in A
- Some probability distributions can be described by **density functions** $p : \mathbb{R}^d \to [0, \infty)$; in this case:

$$\mu(A) = \int_A p(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} \mathbf{1}[x \in A] p(x)\,\mathrm{d}x$$

What is $p(\{(1, 2)\})$? 0 (points have zero probability mass under continuous distributions).
For distributions that do have densities, we often use μ (distribution) and p (its density) interchangeably

Density functions and delta distributions

Do all distributions have density functions?

Density functions and delta distributions

Do all distributions have density functions? No.

Density functions and delta distributions

Do all distributions have density functions? No.

(Dirac) **delta distribution**, or **point mass**, at x: δ_x, defined by:

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}.$$

- δ_x does not have a density function (why?)
- What does this distribution represent? (How do we sample from it?)

Density functions and delta distributions

Do all distributions have density functions? No.

(Dirac) **delta distribution**, or **point mass**, at x: δ_x, defined by:

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}.$$

- δ_x does not have a density function (why?)
- What does this distribution represent? (How do we sample from it?) The random variable $X \sim \delta_x$ is always equal to x.

Density functions and delta distributions

Do all distributions have density functions? No.

(Dirac) **delta distribution**, or **point mass**, at x: δ_x, defined by:

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}.$$

- δ_x does not have a density function (why?)
- What does this distribution represent? (How do we sample from it?) The random variable $X \sim \delta_x$ is always equal to x.
- How do we understand the **empirical distribution**

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i},$$

where $x_1, \ldots, x_n \in \mathbb{R}^d$?

Density functions and delta distributions

Do all distributions have density functions? No.

(Dirac) **delta distribution**, or **point mass**, at x: δ_x, defined by:

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}.$$

- δ_x does not have a density function (why?)
- What does this distribution represent? (How do we sample from it?) The random variable $X \sim \delta_x$ is always equal to x.
- How do we understand the **empirical distribution**

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i},$$

where $x_1, \ldots, x_n \in \mathbb{R}^d$? (Sampling uniformly from $\{x_1, \ldots, x_n\}$.)

Support of a distribution

The **support** of a distribution μ is the smallest closed set S such that $\mu(S) = 1$

- If μ has continuous density p and $p(x) > 0$ for all x, what is the support of μ?
- What is the support of an empirical distribution $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$?
- If $X \sim \text{Uniform}([0,1])$, what is the support of the distribution of $Y = (X, 1-X)$?

Support of a distribution

The **support** of a distribution μ is the smallest closed set S such that $\mu(S) = 1$

- If μ has continuous density p and $p(x) > 0$ for all x, what is the support of μ? The entire $\mathbb{R}^d$. We say μ has **full support**.
- What is the support of an empirical distribution $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$?
- If $X \sim \text{Uniform}([0, 1])$, what is the support of the distribution of $Y = (X, 1 - X)$?

Support of a distribution

The **support** of a distribution μ is the smallest closed set S such that $\mu(S) = 1$

- If μ has continuous density p and $p(x) > 0$ for all x, what is the support of μ? The entire $\mathbb{R}^d$. We say μ has **full support**.
- What is the support of an empirical distribution $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$? $\{x_1, \ldots, x_n\}$.
- If $X \sim \text{Uniform}([0,1])$, what is the support of the distribution of $Y = (X, 1 - X)$?

Support of a distribution

The **support** of a distribution μ is the smallest closed set S such that $\mu(S) = 1$

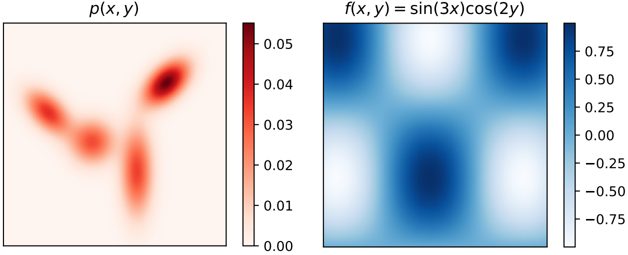
- If μ has continuous density p and $p(x) > 0$ for all x, what is the support of μ? The entire $\mathbb{R}^d$. We say μ has **full support**.
- What is the support of an empirical distribution $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$? $\{x_1, \ldots, x_n\}$.
- If $X \sim \text{Uniform}([0,1])$, what is the support of the distribution of $Y = (X, 1 - X)$? The segment from $(0,1)$ to $(1,0)$; note Y has no density in $\mathbb{R}^2$.

Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p

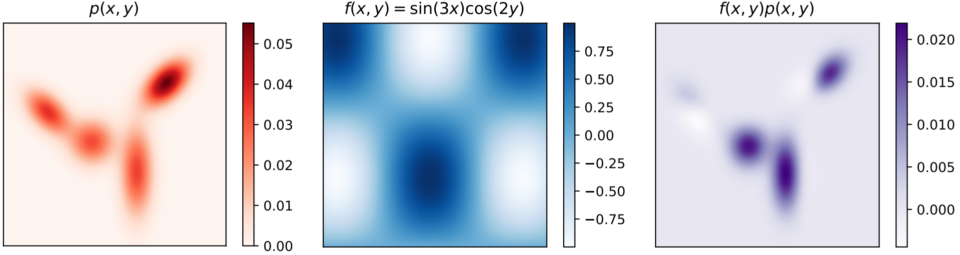


Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p



Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p
- If we sample independently $X_1, \ldots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

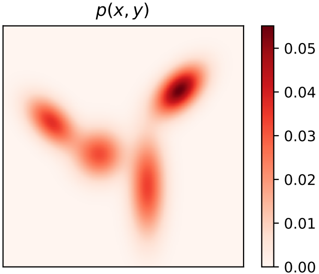
$$\widehat{\mathbb{E}}_{X\sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$

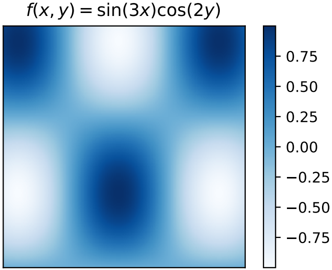
  - This estimator is **unbiased**: $\mathbb{E}[\widehat{\mathbb{E}}_{X\sim p}[f(X)]] = \mathbb{E}_{X\sim p}[f(X)]$
  - **Law of large numbers**: $\widehat{\mathbb{E}}_{X\sim p}[f(X)] \xrightarrow{m\to\infty} \mathbb{E}_{X\sim p}[f(X)]$ (as m increases, the estimate converges to the true value almost surely

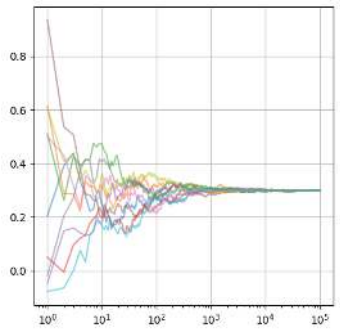
Expectation and Monte Carlo estimation

- If we sample independently $X_1, \ldots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

$$\widehat{\mathbb{E}}_{X\sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$







Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p
- If we sample independently $X_1, \ldots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

$$\widehat{\mathbb{E}}_{X\sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$

- Also for distributions without densities: what is $\mathbb{E}_{X\sim\delta_{x_0}}[f(X)]$?

Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X \sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p
- If we sample independently $X_1, \ldots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

$$\widehat{\mathbb{E}}_{X \sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$

- Also for distributions without densities: what is $\mathbb{E}_{X \sim \delta_{x_0}}[f(X)]$? $f(x_0)$.

Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p
- If we sample independently $X_1, \dots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

$$\widehat{\mathbb{E}}_{X\sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$

- Also for distributions without densities: what is $\mathbb{E}_{X\sim\delta_{x_0}}[f(X)]$? $f(x_0)$.
- What should $\mathbb{E}_{X\sim\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}}[f(X)]$ be?

Expectation and Monte Carlo estimation

- For a distribution with density p, and a function $f : \mathbb{R}^d \to \mathbb{R}$, the **expectation** of $f(X)$ for $X \sim p$ is:

$$\mathbb{E}_{X\sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)p(x)\,\mathrm{d}x$$

  - Could be infinite or undefined for some f and p
- If we sample independently $X_1, \ldots, X_m \sim p$, then the **Monte Carlo estimator** of the expectation is:

$$\widehat{\mathbb{E}}_{X\sim p}[f(X)] = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$$

- Also for distributions without densities: what is $\mathbb{E}_{X\sim\delta_{x_0}}[f(X)]$? $f(x_0)$.
- What should $\mathbb{E}_{X\sim\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}}[f(X)]$ be? $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$.

Generative processes as distributions

Two questions to ask about a distribution used in modelling:

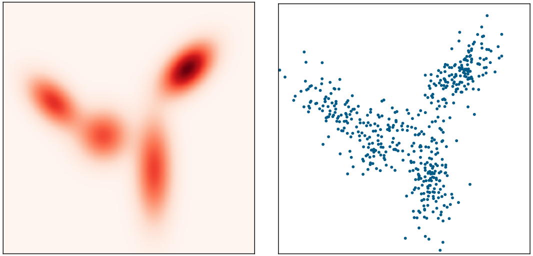
- How to sample from it? (Generative processes are sampling procedures!)
- How to evaluate its density at a given point?

For which of these processes can we evaluate the density?

Generative processes as distributions

For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters.



Generative processes as distributions

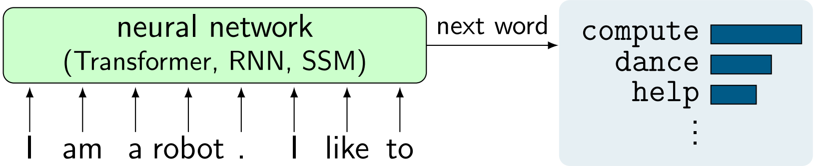
For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset.

Generative processes as distributions

For which of these processes can we evaluate the density?

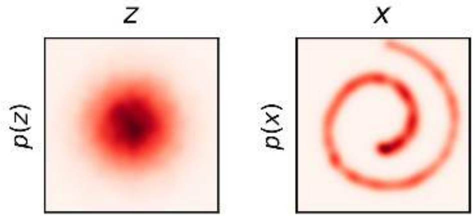
- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset. No density.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence.



Generative processes as distributions

For which of these processes can we evaluate the density?

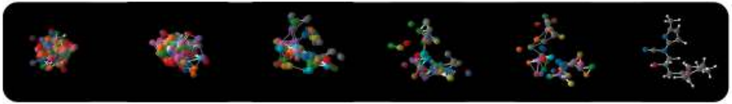
- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset. No density.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until <end> is produced; output the sequence. Not a distribution over $\mathbb{R}^d$, but mass function by autoregressive factorisation.
- Sample $z \sim \mathcal{N}(0, I)$, then output $G(z)$, where G is a neural network.



Generative processes as distributions

For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset. No density.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until <end> is produced; output the sequence. Not a distribution over $\mathbb{R}^d$, but mass function by autoregressive factorisation.
- Sample $z \sim \mathcal{N}(0, I)$, then output $G(z)$, where G is a neural network. No (in general), but more in two weeks.
- Sample a random point cloud and run a physics simulation for a fixed time horizon. Output the resulting point cloud.



Generative processes as distributions

For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset. No density.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence. Not a distribution over $\mathbb{R}^d$, but mass function by autoregressive factorisation.
- Sample $z \sim \mathcal{N}(0, I)$, then output $G(z)$, where G is a neural network. No (in general), but more in two weeks.
- Sample a random point cloud and run a physics simulation for a fixed time horizon. Output the resulting point cloud. No (in general), but more at the end of the track.

Generative processes as distributions

For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters. Yes.
- Sample a random point from the dataset. No density.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence. Not a distribution over $\mathbb{R}^d$, but mass function by autoregressive factorisation.
- Sample $z \sim \mathcal{N}(0, I)$, then output $G(z)$, where G is a neural network. No (in general), but more in two weeks.
- Sample a random point cloud and run a physics simulation for a fixed time horizon. Output the resulting point cloud. No (in general), but more at the end of the track.

Are there distributions for which we can evaluate the density, but not (easily) sample from them?

Generative processes as distributions

For which of these processes can we evaluate the density?

- Sample from a Gaussian mixture with known parameters.
- Sample a random point from the dataset.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence.
- Sample $z \sim \mathcal{N}(0, I)$, then output $G(z)$, where G is a neural network.
- Sample a random point cloud and run a physics simulation for a fixed time horizon. Output the resulting point cloud.

Are there distributions for which we can evaluate the density, but not (easily) sample from them? Yes: Bayesian posteriors $p(x \mid y) \propto p(x)p(y \mid x)$, for example. Many methods exist to sample approximately.

- Some notes and review of Lecture 1
- Preliminaries
  - Probability distributions and density functions
  - Generative processes
- Generative modelling as an optimisation problem
  - Divergence measures

Generative modelling as distribution approximation

Setting:

- We have a **data distribution** π_{data} over $\mathbb{R}^d$ (from which we can sample, but we do not know its density function)
  - It could be the empirical distribution of a dataset
- We have a class of **model distributions** $\{\pi_\theta\}$ (with densities p_θ)
  - θ are the parameters of the model (*e.g.*, neural network weights, Gaussian mixture parameters)
  - Note that we do not necessarily know the density functions p_θ
- We seek θ such that π_θ approximates π_{data} well:

$$\theta^* = \arg\min_\theta D(\pi_\theta, \pi_{\text{data}})$$

  - Next: What is D?

- Some notes and review of Lecture 1
- Preliminaries
  - Probability distributions and density functions
  - Generative processes
- Generative modelling as an optimisation problem
  - Divergence measures

What should a divergence measure be?

Some desirable properties:

What should a divergence measure be?

Some desirable properties:

- Nonnegativity: $D(\pi_{\text{data}}, \pi_{\text{model}}) \geq 0$, with equality only if $\pi_{\text{data}} = \pi_{\text{model}}$
- Easy estimation from samples
- Optimisation tractability
  - Some measures (*e.g.*, transport-based) are good for model evaluation, but not for training (more on this in a few weeks)

Kullback-Leibler divergence

If p and q are (densities of) two distributions, the **Kullback-Leibler (KL) divergence** from p to q is defined as:

$$\mathrm{KL}(p \| q) = \mathbb{E}_{X \sim p}\left[\log \frac{p(X)}{q(X)}\right]$$

Kullback-Leibler divergence

If p and q are (densities of) two distributions, the **Kullback-Leibler (KL) divergence** from p to q is defined as:

$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$$

Kullback-Leibler divergence

If p and q are (densities of) two distributions, the **Kullback-Leibler (KL) divergence** from p to q is defined as:

$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x$$

- **Gibbs' inequality:** $\mathrm{KL}(p\|q) \geq 0$, equality only if $p = q$ as distributions
- Importantly, $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$ in general

Kullback-Leibler divergence

If p and q are (densities of) two distributions, the **Kullback-Leibler (KL) divergence** from p to q is defined as:

$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)}\, \mathrm{d}x$$

- **Gibbs' inequality:** $\mathrm{KL}(p\|q) \geq 0$, equality only if $p = q$ as distributions
- Importantly, $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$ in general
- When/how can the KL be estimated using samples?

Kullback-Leibler divergence

If p and q are (densities of) two distributions, the **Kullback-Leibler (KL) divergence** from p to q is defined as:

$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)}\, \mathrm{d}x$$

- **Gibbs' inequality:** $\mathrm{KL}(p\|q) \geq 0$, equality only if $p = q$ as distributions
- Importantly, $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$ in general
- When/how can the KL be estimated using samples? If we can sample from p and evaluate both densities, use Monte Carlo.

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\text{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\text{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\text{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{"forward" KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\text{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{"reverse" KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

However, $\text{KL}(\pi_{\text{data}} \| \pi_\theta)$ is more suitable, because:

$$\text{KL}(\pi_{\text{data}} \| \pi_\theta) = \mathbb{E}_{X \sim \pi_{\text{data}}} \left[\log \frac{p_{\text{data}}(X)}{p_\theta(X)} \right]$$

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

However, $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ is more suitable, because:

$$\begin{aligned} \mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) &= \mathbb{E}_{X \sim \pi_{\text{data}}}\left[\log \frac{p_{\text{data}}(X)}{p_\theta(X)}\right] \\ &= \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_{\text{data}}(X)]} - \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]} \end{aligned}$$

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

However, $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ is more suitable, because:

$$\begin{aligned} \mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) &= \mathbb{E}_{X \sim \pi_{\text{data}}}\left[\log \frac{p_{\text{data}}(X)}{p_\theta(X)}\right] \\ &= \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_{\text{data}}(X)]}_{\substack{\text{some unknown constant} \\ \text{(negative entropy)}}} - \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]}_{} \end{aligned}$$

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

However, $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ is more suitable, because:

$$\begin{aligned}
\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) &= \mathbb{E}_{X \sim \pi_{\text{data}}}\left[\log \frac{p_{\text{data}}(X)}{p_\theta(X)}\right] \\
&= \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_{\text{data}}(X)]}_{\substack{\text{some unknown constant} \\ \text{(negative entropy)}}} - \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]}_{\text{can be estimated from samples}}
\end{aligned}$$

Using KL divergence for generative modelling

Which direction to use for generative modelling (given samples from π_{data})?

$$\theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)}^{\text{“forward” KL}} \quad \text{or} \quad \theta^* = \arg\min_\theta \overbrace{\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})}^{\text{“reverse” KL}}?$$

If we do not have the density of π_{data}, we can compute **neither** directly!

However, $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ is more suitable, because:

$$\begin{aligned} \mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) &= \mathbb{E}_{X \sim \pi_{\text{data}}}\left[\log \frac{p_{\text{data}}(X)}{p_\theta(X)}\right] \\ &= \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_{\text{data}}(X)]}_{\substack{\text{some unknown constant} \\ \text{(negative entropy)}}} - \underbrace{\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]}_{\text{can be estimated from samples}} \end{aligned}$$

Minimising $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]$

Maximum likelihood estimation

Minimising $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]$

- Recovers **maximum likelihood estimation** (MLE)
  - Maximising joint probability $\log \prod_{x_i \in \text{dataset}} p_\theta(x_i)$
  - Assumes π_{data} is the distribution of independent samples from an underlying distribution

Maximum likelihood estimation

Minimising $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]$

- Recovers **maximum likelihood estimation** (MLE)
  - Maximising joint probability $\log \prod_{x_i \in \text{dataset}} p_\theta(x_i)$
  - Assumes π_{data} is the distribution of independent samples from an underlying distribution
- If we can compute $p_\theta(x)$ for any x, and draw samples $x \sim \pi_{\text{data}}$, we can estimate this expectation using Monte Carlo:

$$\widehat{\mathbb{E}}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)] = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i), \quad x_i \sim \pi_{\text{data}}$$

Maximum likelihood estimation

Minimising $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]$

- If we can compute $p_\theta(x)$ for any x, and draw samples $x \sim \pi_{\text{data}}$, we can estimate this expectation using Monte Carlo:

$$\widehat{\mathbb{E}}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)] = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i), \quad x_i \sim \pi_{\text{data}}$$

- Algorithm to fit θ using stochastic gradient descent
  - Sample a minibatch $x_1, \ldots, x_m \sim \pi_{\text{data}}$
  - Compute gradient estimate:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta [-\log p_\theta(x_i)]$$

  - Update parameters: $\theta \leftarrow \theta - \eta g$ (or using your favourite optimiser)

Maximum likelihood estimation

Minimising $\mathrm{KL}(\pi_{\mathrm{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\mathrm{data}}}[\log p_\theta(X)]$

- Algorithm to fit θ using stochastic gradient descent
  - Sample a minibatch $x_1, \ldots, x_m \sim \pi_{\mathrm{data}}$
  - Compute gradient estimate:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta [-\log p_\theta(x_i)]$$

  - Update parameters: $\theta \leftarrow \theta - \eta g$ (or using your favourite optimiser)
- What does this algorithm require?

Maximum likelihood estimation

Minimising $\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X \sim \pi_{\text{data}}}[\log p_\theta(X)]$

- Algorithm to fit θ using stochastic gradient descent
  - Sample a minibatch $x_1, \ldots, x_m \sim \pi_{\text{data}}$
  - Compute gradient estimate:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta [-\log p_\theta(x_i)]$$

  - Update parameters: $\theta \leftarrow \theta - \eta g$ (or using your favourite optimiser)
- What does this algorithm require? p_θ known and differentiable w.r.t. θ.

Jensen-Shannon divergence

A compromise: the **Jensen–Shannon (JS) divergence**

$$\mathrm{JS}(p, q) = \frac{1}{2}\mathrm{KL}\left(p \middle\| \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(q \middle\| \frac{p+q}{2}\right)$$

- $\mathrm{JS}(p, q) \geq 0$, with equality only if $p = q$ as distributions
- $\mathrm{JS}(p, q) = \mathrm{JS}(q, p)$
- $0 \leq \mathrm{JS}(p, q) \leq \log 2$ (or ≤ 1, if using base-2 log)

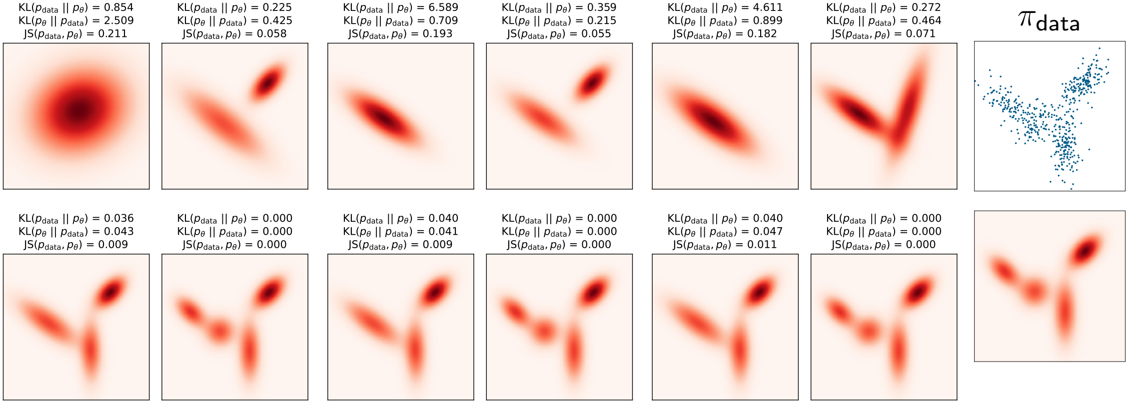
Summary of three divergences considered

Which divergence to use for generative modelling, if all are possible?

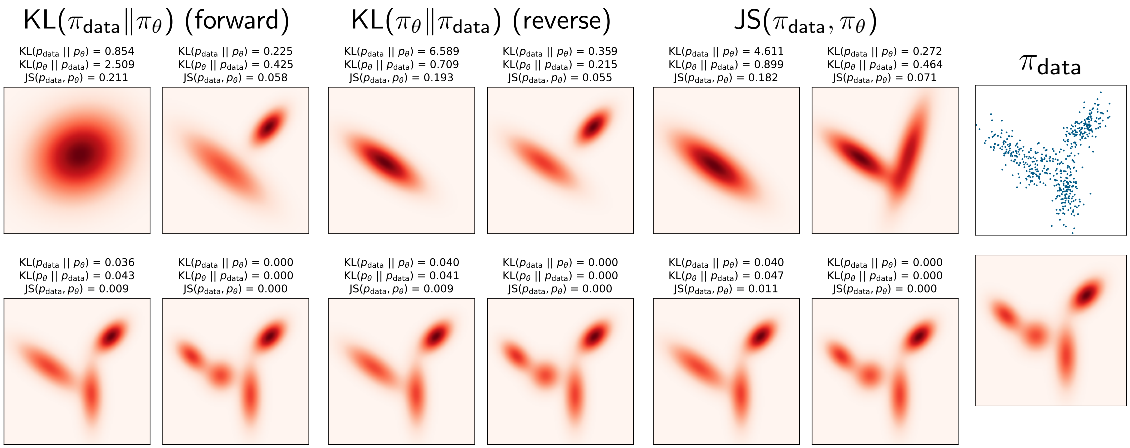
Summary of three divergences considered

Which divergence to use for generative modelling, if all are possible?

$\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ (forward) $\quad$ $\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})$ (reverse) $\quad$ $\mathrm{JS}(\pi_{\text{data}}, \pi_\theta)$



Summary of three divergences considered



- Forward KL / MLE: **mode-covering** (high diversity, low fidelity)
- Reverse KL: **mode-seeking** (high fidelity, low diversity)

Conclusion and looking ahead

- Generative modelling can be formulated as optimisation of a divergence between the data distribution and model distribution
- Forward KL divergence minimisation $\equiv$ maximum likelihood estimation
- Tutorial: exploring choices of divergence for fitting simple models
- Next time: latent variable models (when p_θ not available in closed form) and autoencoders
  - Suggestion to review variational inference from PMR course or Probabilistic ML book (Advanced Topics, §10.1-2) for advanced reading