Lecture Notes

PROBABILISTIC GENERATIVE MODELS

This version: 22nd January 2026

Latest version: github.com/felipe-tobar/Probabilistic-Generative-Models

Felipe Tobar
Department of Mathematics
Imperial College London

f.tobar@imperial.ac.uk
https://www.ma.ic.ac.uk/~ft410/

Preface

This notes are under development for 2026.

Felipe Tobar,
London,
January 2026.

Contents

Week 1

Foundations

1.1 Introduction

A Probabilistic generative model (PGM), or simply, a GM, is a methodology for generating data. In general, the PGM is constructed and adjusted using observations with the aim to synthesise samples with the same statistical properties of the available observations. The *probabilistic* nature of the PGMs studied in this course follows from the fact that the available data will be considered to be realisations of an underlying random variable (RV), e.g., X.

In this sense, the probability distribution of X, denoted $P_X(x)$, as well as its probability density function (pdf) $p_X(x)$ will be central to the study of PGMs. In particular, targetting the pdf is one way of constructing PGMs, in which case the whole PGM paradigm becomes equivalent to the classical statistical modelling approach. However, as we will see in the course, enforcing the sought-after PGM to have an explicit parametric pdf can be rather restrictive.

Throughout the course, we will consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with 3 RVs given by the following measurable maps:

$$\underset{\text{(observed input)}}{X : \Omega \to \mathcal{X}} \qquad \underset{\text{(observed output)}}{Y : \Omega \to \mathcal{Y}} \qquad \underset{\text{(latent variable)}}{Z : \Omega \to \mathcal{Z}}$$

Remark 1.1.

Not all three RVs will be present in all our settings. For instance, in classification there is no justification for the latent variable Z (in general), while in clustering, there is no need for X. However, we build the general setup here for formality.

We equip $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ with their Borel σ-algebras $\mathcal{B}(\mathcal{X}), \mathcal{B}(\mathcal{Y}), \mathcal{B}(\mathcal{Z})$, and consider the product measurable space

$$(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z},\ \mathcal{B}(\mathcal{X}) \otimes \mathcal{B}(\mathcal{Y}) \otimes \mathcal{B}(\mathcal{Z})).$$

Then the joint random variable $(X, Y, Z) : \Omega \to \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ is measurable with respect to $\mathcal{F}$ and the above product σ-algebra.

Furthermore, we will assume that the law of (X, Y, Z) is absolutely continuous with respect to the product base measure on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ (e.g. Lebesgue measure when $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \mathbb{R}^d$), and hence admits a joint density $p(x, y, z)$. That is, for all $A \in \mathcal{B}(\mathcal{X}), B \in \mathcal{B}(\mathcal{Y}), C \in$

$\mathcal{B}(\mathcal{Z})$,

$$\mathbb{P}\left(X \in A, Y \in B, Z \in C\right) = \int_{A\times B\times C} p(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z. \tag{1.1}$$

We will also assume that all relevant marginals and conditionals admit densities (with respect to the corresponding base measures), e.g. $p(x,y)$, $p(y|x)$, $p(z|x,y)$, etc.

1.2 Discriminative versus generative

The generative approach aims to characterise the complete generative distribution $p(x,y,z)$, whereas, in some application-specific cases, only the discriminative model, e.g., $p(y|x)$, is needed. Let us examine the following example.

Example 1.1 (Generative and discriminative views of binary classification)**.**
Consider the binary classification problem, where, given an observation $X = x$, one needs to estimate its label Y. A discriminative model would directly parametrise $\mathbb{P}(Y|X = x)$. Since this is a binary classification case, without loss of generality, we can assume $Y \in \{0,1\}$, and model $\mathbb{P}(Y = 1|X = x)$, since $\mathbb{P}(Y = 0|X = x) = 1 - \mathbb{P}(Y = 1|X = x)$. A model for this probability only needs to map $x \in \mathbb{R}^d \to \mathbb{P}(Y = 1|X = x) \in [0,1]$. For instance, a reasonable candidate for this is

$$\mathbb{P}(Y = 1|X = x) = \frac{1}{1 + e^{-\theta^\top x}} \tag{1.2}$$

which is known as the logistic regression.

Conversely, in a generative approach, we aim to model the joint probability $p(Y = y, X = x)$. Modelling this distribution is not easy, however, observe that we can factorise it as

$$p(Y = y, X = x) = p(X = x|Y = y)p(Y = y), \tag{1.3}$$

which yields a pair of much more intuitive distributions to model:

- the class probability $p(Y = y) = (\pi, 1-\pi), \pi \in [0,1]$, and
- the class-conditional probability $p(X = x|Y = y)$, given by a two distributions over $\mathcal{X}$, denoted f_{θ_0} and f_{θ_1}.

Therefore, the classifier is

$$\begin{aligned} p(Y = 1|X = x) &= \frac{p(X = x|Y = 1)p(Y = 1)}{p(X = x)} \\ &= \frac{1}{1 + e^{-\log\left(\frac{\pi}{1-\pi}\frac{f_{\theta_1}(x)}{f_{\theta_0}(x)}\right)}}. \end{aligned} \tag{1.4}$$

Exercise 1.1.
Evaluate eq. (1.4) for $f_{\theta_0} = \mathcal{N}(\mu_0, \Sigma_0)$ and $f_{\theta_1} = \mathcal{N}(\mu_1, \Sigma_1)$. What happens when $\Sigma_0 = \Sigma_1$?

1.3 The pushforward measure

Despite the abundant collection of well-studied statistical models, in some scenarios we can construct a more ad hoc model by applying an appropriate transformation.

Definition 1.1.
Consider a RV $X \in \mathcal{X}$ with measure P_X, and a nonlinear map $T : \mathcal{X} \to \mathcal{X}$. The measure of the transformed RV $T(X)$ is known as the *push forward measure* of P_X through T, and it is denoted by $T_{\#}P_X$

Remark 1.2.
The transformations considered in the course will be such that the pushforward measure has a density. With a slight abuse of notation, we will denote this density as $T_{\#}p_X$.

Example 1.2 (Discrete pushforward)**.**
Let X be a discrete random variable taking values in $\{1, 2, 3\}$ with

$$\mathbb{P}(X = 1) = 0.2, \quad \mathbb{P}(X = 2) = 0.5, \quad \mathbb{P}(X = 3) = 0.3.$$

Define the map $T : \{1, 2, 3\} \to \{a, b\}$ by

$$T(1) = a, \qquad T(2) = b, \qquad T(3) = b.$$

Then the pushforward $T_{\#}\mathbb{P}$ satisfies

$$(T_{\#}\mathbb{P})(\{a\}) = 0.2, \qquad (T_{\#}\mathbb{P})(\{b\}) = 0.8.$$

For an illustration see Fig. 1.1.

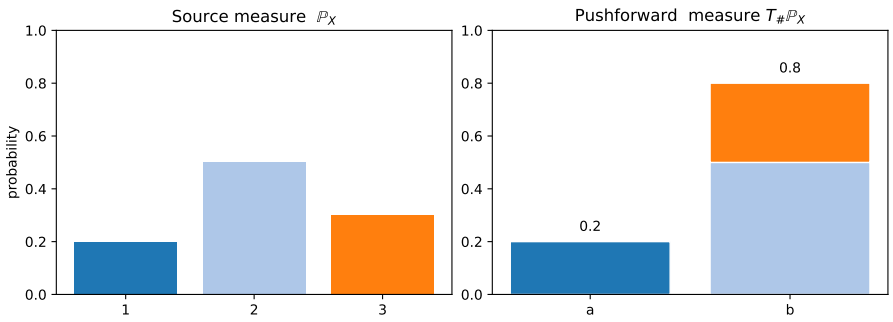

Figure 1.1: Source and pushforward distributions: discrete example.

Example 1.3 (Continuous pushforward)**.**
Let $X \sim \mathcal{N}(0, 1)$ on $\mathbb{R}$ and define $T(x) = x^2$. The pushforward $T_{\#}\mathbb{P}$ is the law of $Y = T(X)$, supported on $\mathbb{R}_+$. Its density is given by

$$p_Y(y) = \frac{1}{\sqrt{2\pi y}} \exp\left(-\frac{y}{2}\right), \qquad y > 0.$$

For an illustration of this case and other related examples, see Fig. 1.2.

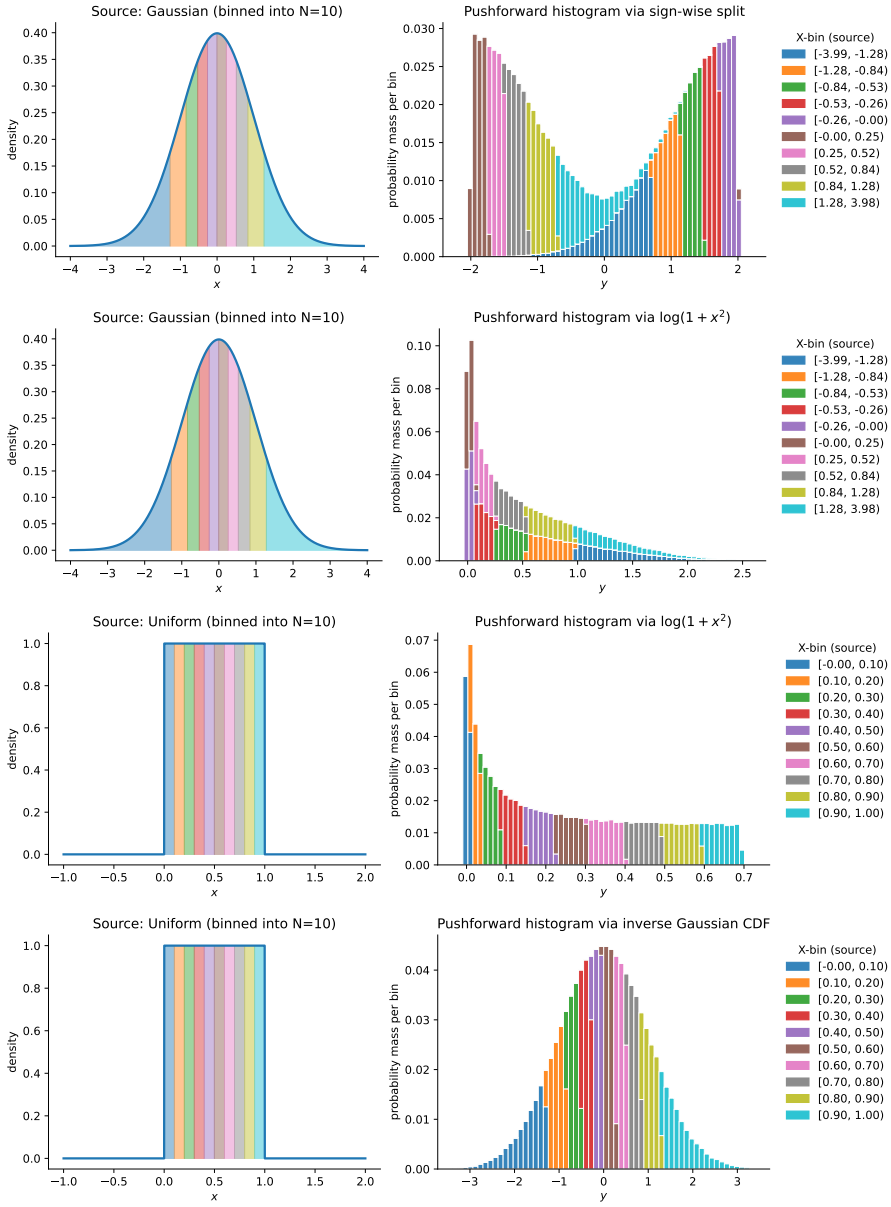

Figure 1.2: Source and target distributions: continuous examples

In general, for arbitrary source distributions and maps it is difficult to compute the target density in closed form, at least in the continuous case. For the specific case of differentiable and invertible maps T, the following theorem given a recipe to compute $p_{T(X)}$

Theorem 1.1 (Change of variable)**.**
Consider two RVs $X, Y \in \mathbb{R}^d$, such that $Y = T(X)$, where $T : \mathbb{R}^d \to \mathbb{R}^d$ is a C^1 diffeomorphism. If X and Y have densities p_X and p_Y respectively, then

$$p_Y(y) = p_X(T^{-1}(y)) \left| \det \nabla_y T^{-1}(y) \right|, \tag{1.5}$$

where $\nabla_y T^{-1}(y)$ is the Jacobian of the inverse map.

Remark 1.3.

Though the above result provides a closed-form expression for the pushforward measure only when T is a C^1 diffeomorphism (continuously differentiable with an the inverse having the same property), we can transform a source RV X into a target RV T with any measurable map. This is because

$$(T_{\#}P_X)(A) \;=\; P_X(T^{-1}(A)), \qquad \forall A \in \mathcal{B}(\mathcal{X}). \tag{1.6}$$

Though in general the pdf of T will not be available in closed form.

1.4 Likelihood-based training

Maximum likelihood (ML) is going to be the canonical methodology for training our PGMs, and, as we will see next, it will recover other forms of training criteria in particular cases.

Consider a PGM for the RV Y, with density $p_\theta(y)$, where $\theta \in \Theta$ denotes the model parameter. Also, consider the realisations of Y given by $y_1, y_2, \ldots, y_N$.

Definition 1.2 (Likelihood function)**.**
The likelihood of the parameter θ is the function $L : \Theta \to \mathbb{R}_+$ given by the probability density function of Y evaluated on the observations. That is,

$$L(\theta) = p_\theta(y_1, y_2, \ldots, y_N). \tag{1.7}$$

NB: We abused notation above stating the joint pdf for the observations.

Remark 1.4.
Very important: the likelihood function is not a probability/density function, as it is a function of the parameter. In particular, it is not true that $\int_\Theta L(\theta)\mathrm{d}\theta$ is one.

Definition 1.3 (Maximum likelihood estimator)**.**
The ML estimator is given by

$$\theta_{ML} = \arg\max L(\theta). \tag{1.8}$$

Remark 1.5.
In general (but, importantly, not always) we will consider i.i.d observations, in which case the likelihood factorises as $L(\theta) = \prod_{n=1}^N p_\theta(y_n)$. Furthermore, when optimising the likelihood we will consider the log-likelihood instead; in the i.i.d. case, this is

$$l(\theta) = \log L(\theta) = \sum_{n=1}^{N} \log p_\theta(y_n). \tag{1.9}$$

Example 1.4 (Gaussian linear regression)**.**
Let us consider the PGM given by

$$Y|x \sim \mathcal{N}(ax, \sigma^2), a, x \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+. \tag{1.10}$$

This is equivalent to $Y = ax + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$. The parameters in this setting are $\theta = (a, \sigma^2)$. Now consider the observations $\{(x_n, y_n)\}_{n=1}^N$.

Since $p(y_n|x_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(y_n - ax_n)^2\right)$, we can write the log-likelihood as

$$l(\theta) = \sum_{n=1}^{N} \frac{-1}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y_n - ax_n)^2. \tag{1.11}$$

The optimal (a, σ^2) can be found in closed form using the first order optimality conditions.

Remark 1.6.
Observe that optimising eq. (1.11) recovers the least squares solution.

Example 1.5 (Binary classification)**.**
Consider observations $\{(x_n, y_n)\}_{n=1}^N \subset \mathbb{R}^d \times \{0, 1\}$ from a binary classification setting. Model the classifier as

$$p_\theta(y = 1|x) = \sigma(s(x)), \tag{1.12}$$

where $\sigma(s(x)) = \frac{1}{1+e^{-s(x)}}$, and $s : \mathbb{R}^d \to \mathbb{R}$ is a feature extractor (e.g., $s(x) = a^\top x + b$). Assuming that the observations are i.i.d., we have

$$L(\theta) = \prod_{n=1}^{N} p(y_n|x_n) = \prod_{n=1}^{N} \sigma(s(x_n))^{y_n}(1 - \sigma(s(x_n)))^{1-y_n}, \tag{1.13}$$

and equivalently

$$l(\theta) = \sum_{n=1}^{N} y_n \log \sigma(s(x_n)) + (1 - y_n) \log(1 - \sigma(s(x_n))). \tag{1.14}$$

Does this expression seem familiar? If not, we will find out soon what this is.

Example 1.6 (Clustering)**.**
Consider a set of observations $\{x_n\}_{n=1}^N \in \mathbb{R}^d$ and implement a clustering algorithm. We will assume that there are $K \in \mathbb{N}$ clusters, each specified by a density $p_k, k = 1, \ldots, K$; this means that the probability of the RV X coming from the k-th cluster is $\mathbb{P}(X \in C_k) = \pi_k$, where $\forall k, 0 \leq \pi_k \leq 1$ and $\sum_{k=1}^K \pi_k = 1$.

This is a mixture model, with density $p(x) = \sum_{k=1}^K \pi_k p_k(x)$, and parameters given by the cluster probabilities π_k and the parameters of the densities $p_k = p_{\theta_k}$. The log-likelihood is

$$l(\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p_k(x_n) \tag{1.15}$$

Note that there are two issues associated to optimising eq. (1.15).

- We do not recover the cluster assignments.
- The problem is ill-posed. E.g., if $p_k = \mathcal{N}(\mu_k, \Sigma_k)$, which is the usual choice, we can set $\mu_k = x_n, \Sigma_k = 0$ which gives $l = \infty$.

We can overcome this drawback by introducing a collection of latent random variables $Z_{nk} \in \{0, 1\}$, that represents the cluster assignments. That is,

$$Z_{nk} = 1 \iff x_n \in C_k. \tag{1.16}$$

This allows us to write the conditional densities $p(x_n|z_{nk}) = \prod_{k=1}^{K} p_k^{z_{nk}}$, and thus to express the **complete-data likelihood** given by

$$l(\theta) = \log \prod_{n=1}^{N} \prod_{k=1}^{K} p_k^{z_{nk}}(x_n) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log p_k(x_n). \tag{1.17}$$

Good and bad news: this objective is now theoretically feasible to optimise but impractical since we do not have access to the latent cluster assignments $\{z_{nk}\}_{nk}$.

A workaround to this is to estimate the cluster assignments, via its conditional expectation wrt the observations. That is,

$$\mathbb{E}\left(z_{nk}|x_{1:N}\right) = 1 * \mathbb{P}(z_{nk} = 1|x_n) + 0 * \mathbb{P}(z_{nk} = 0|x_n) = \mathbb{P}(z_{nk} = 1|x_n), \tag{1.18}$$

which can be computed explicitly using Bayes theorem in terms of the model parameters. Then, we can perform and iterative procedure by: i) optimising $l(\theta)$ using $\mathbb{E}\left(z_{nk}|x_{1:N}\right)$, and ii) computing $\mathbb{E}\left(z_{nk}|x_{1:N}\right)$ using $\theta_{ML} = \arg\max l(\theta)$.

This means that exact ML cannot be performed in this case. Also, does this procedure seem familiar?

The maximum likelihood estimator (MLE) satisfies several important theoretical properties:

- **Consistency:** Under the assumption that the statistical model is *identifiable*—i.e., different parameter values correspond to different probability distributions—the MLE converges to the true parameter as the number of observations grows. Intuitively, maximising the likelihood asymptotically minimises the Kullback–Leibler divergence between the true distribution and the distribution induced by a candidate parameter.

- **Equivariance:** If $\hat{\theta}_{\text{MLE}}$ is the MLE of θ, then for any transformation g, the MLE of $g(\theta)$ is $g(\hat{\theta}_{\text{MLE}})$. This property allows us to compute MLEs under reparametrisations directly.

- **Asymptotic normality:** For large sample sizes, the MLE is approximately normally distributed around the true parameter with covariance matrix given by the inverse Fisher information. Formally,

$$\sqrt{n}(\hat{\theta}_{\text{MLE}} - \theta) \xrightarrow{d} \mathcal{N}(0, I(\theta)^{-1}),$$

where $I(\theta)$ is the Fisher information matrix.

- **Asymptotic efficiency:** As a consequence of asymptotic normality, the MLE achieves the Cramér–Rao lower bound for the variance in the limit of large n, making it asymptotically optimal among unbiased estimators.

In practice, these properties justify the widespread use of the MLE: it not only converges to the true parameter under mild assumptions, but also allows for straightforward reparametrisations and provides an estimator with minimal asymptotic variance.

1.5 A brief intro to information theory

NB: This section is based on Chapter 6 of (Murphy, 2022)

Motivation. Let us consider a discrete RV $X \in \{1, 2, \dots, K\}$ with pmf p_X. Observe that $-\log p_X(x)$ represents a measure of *information* gained from obtaining the value a as a sample of X. Now consider a communication channel $A \to B$, where A is transmitting samples of X to B. When B received the samples, its *average information* can be expressed as

$$H(X) = -\sum_{x=1}^{K} p_X(x) \log p_X(x). \tag{1.19}$$

This quantity is known as *entropy* and—in connection with thermodynamics—it represent a measure of disorder or un-predictability of X.

NB: We will use $H(X)$ and $H(p_X)$ interchangeably.
NB: We will usually denote $H(X) = -\mathbb{E}\left(\log p_X\right) = \mathbb{E}\left(\log \frac{1}{p_X}\right)$.

Clearly, $H(X) \geq 0$ with equality achieved for an RV that has always the same outcome with $p_X(x) = 1$. This is the deterministic, predictable, case. To revise further properties, let us recall the following result.

Jensen's Inequality Let $\phi : \mathbb{R} \to \mathbb{R}$ be a *convex* function and let X be a random variable such that $\mathbb{E}[|X|] < \infty$. Then

$$\phi\Big(\mathbb{E}[X]\Big) \;\leq\; \mathbb{E}\Big[\phi(X)\Big]. \tag{1.20}$$

since $\log(\cdots)$ is *concave*, the inequality is reversed and we have:

$$\mathbb{E}[\log X] \;\leq\; \log \mathbb{E}[X]. \tag{1.21}$$

Remark 1.7.

Equality in eq. (1.20) is only achieved when either ϕ is affine, or X is constant almost surely, that is, $\mathbb{P}\left(X = c\right) = 1$. As a consequence, equality in eq. (1.21) is only achieved when X is constant almost surely (when the argument of the logarithm does not depend on x)

Keep this result in mind, as it will be used throughout the module.

Using Jensen on the definition of the entropy, we have

$$H(X) = \mathbb{E}\left(\log \frac{1}{p}\right) \leq \log \mathbb{E}\left(\frac{1}{p}\right) = \log \sum \frac{1}{p} p = \log K. \tag{1.22}$$

This directly implies that the uniform distribution over $\{1, 2, \ldots, K\}$ has the largest entropy, since

$$H(U_{1:K}) = \mathbb{E}\left(\log \frac{1}{1/K}\right) = \log K. \tag{1.23}$$

Example 1.7 (Bernoulli distribution)**.**
Consider $X \sim p_X(x) = \theta^x(1-\theta)^{1-x}, \theta \in [0, 1]$. The entropy is given by $H(X) = -\theta \log \theta - (1-\theta)\log(1-\theta)$. Figure 1.3 shows this function.

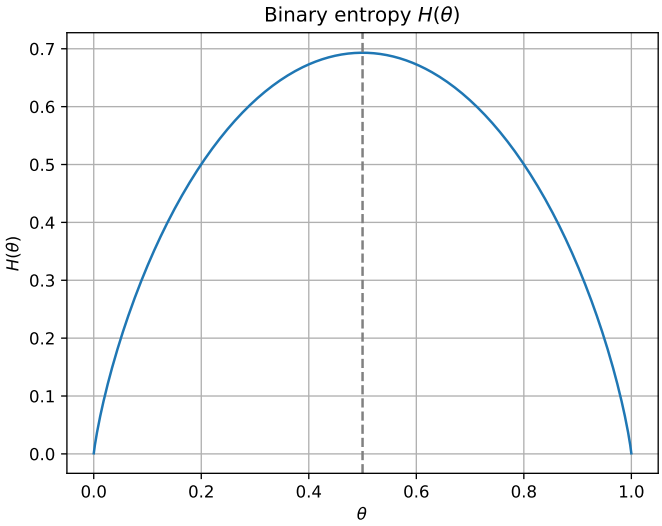

Figure 1.3: Entropy for Bernoulli.

The entropy, in addition to being a measure of disorder, can be understood as the cost of the optimal for of compression. Here, think of a compression strategy using symbols $s_1, s_2, \ldots$ with increasing size (or storage cost). For instance, think of storage using logical gates, meaning that these symbols are (equivalent to)

$$s_1 = 0, s_2 = 1, s_3 = 10, s_4 = 11, \ldots \tag{1.24}$$

where storing more and more symbols becomes increasingly expensive. The compression strategy is then to assign each outcome of X with a symbol s_i. Intuitively, the optimal compression would assign s_i to the i-th most frequent value in $\{1, 2, \ldots., K\}$, meaning that the *cost of storage* precisely grows precisely with $\log \frac{1}{p_X}$.

As a consequence, the average message size is the entropy $H(X)$, and thus can be understood as the cost of this compression strategy.

Now let us go back to our communication channel $A \to B$, where the receiver in B now mistakenly believes that $X \sim q_X$. B's estimated entropy, or averaged information, would be

$$H_{CE}(p_X, q_X) = -\sum_{x=1}^{K} p_X(x) \log q_X(x). \tag{1.25}$$

Following the same rationale as above, this can be interpreted as the cost of compressing the sequence $x_1, x_2, x_3, \ldots$ using q_x.

This quantity is known as the cross-entropy between p_X and q_X. Note that this quantity is not symmetric.

It is relevant to study how $H_{CE}(p_X, q_X)$ and $H(P) = H_{CE}(p_X, p_X)$ relate to one another, in particular if one of them is (always) larger than the other.

Let us see:

$$H(p) - H_{CE}(p_X, q_X) = \sum_x p(x) \log \frac{q(x)}{p(x)} \tag{1.26}$$

$$\overset{\text{Jensen's}}{\leq} \log \sum_x \cancel{p(x)} \cancel{\frac{q(x)}{p(x)}} \tag{1.27}$$

$$= \log 1 = 0. \tag{1.28}$$

Therefore, $H(p) \leq H(p, q)$, with equality only achieved when $p = q$, as per Remark 1.7.

Remark 1.8.
Minimising the cross-entropy wrt to one of its arguments is precisely an attempt to match $p = q$.

The use of the entropy/cross-entropy that is going to be more relevant in our case is via its application to continuous RVs. This extension is

$$H(p) = -\int_{\mathcal{X}} p(x) \log p(x) \mathrm{d}x \tag{1.29}$$

$$H(p, q) = -\int_{\mathcal{X}} p(x) \log q(x) \mathrm{d}x \tag{1.30}$$

$$\tag{1.31}$$

Remark 1.9.
Unlike its discrete formulation, $H(p)$ can be positive, negative or zero for continuous RVs.

Example 1.8 (Continuous unifom distribution)**.**
Consider a RV $X \sim U_{[a,b]}$, its entropy is

$$H(U_{[a,b]}) = -\int_a^b \frac{1}{b-a} \log \frac{1}{b-a} \mathrm{d}x = \log(b-a).$$

and can be zero (resp. negative) if $b - a = 1$ (resp. $b - a \leq 1$).

The difference between the entropy and crossentropy is of critical relevance in this module (and life). We will recall a relevant definition first.

Definition 1.4 (Absolute Continuity)**.**
Let $(\Omega, \mathcal{F})$ be a measurable space and let P and Q be probability measures on it. We say that P is *absolutely continuous* with respect to Q, denoted $P \ll Q$, if for every $A \in \mathcal{F}$,

$$Q(A) = 0 \implies P(A) = 0.$$

Remark 1.10.

If $P \ll Q$, and Q admits a density q, P admits a density p satisfying $p(x) = 0$ whenever $q(x) = 0$.

Definition 1.5 (Kullback–Leibler Divergence)**.**
Let P and Q be probability measures on a measurable space $(\Omega, \mathcal{F})$ such that $P \ll Q$. If P and Q admit densities p and q with respect to a common base measure (e.g. Lebesgue measure), then

$$\mathrm{KL}(p \,\|\, q) \;=\; \mathbb{E}_p\Big[\log \frac{p(X)}{q(X)}\Big] \;=\; \int p(x)\, \log \frac{p(x)}{q(x)}\, \mathrm{d}x.$$

Definition 1.6 (Discrete KL Divergence)**.**
For discrete distributions,

$$\mathrm{KL}(p \,\|\, q) \;=\; \sum_x p(x)\, \log \frac{p(x)}{q(x)}.$$

Remark 1.11.
Notice that the KL divergence is always positive:

$$\mathrm{KL}(p \,\|\, q) = H(p, q) - H(p) \geq 0. \tag{1.32}$$

The KL is a *divergence*, i.e., a function that quantifies how far p is form q that is i) always positive, and ii) $\mathrm{KL}(p \,\|\, q) = 0 \iff p = q$ (identify of the indiscernible). However, note that the KL is not a distance, since

- is not symmetric
- does not have triangle inequality.

Critically, the $\mathrm{KL}(p \,\|\, q)$ is only defined when $P \ll Q$.

1.6 KL divergence as a metric to compare p and q

In the continuous case, we are interested in understanding what type of convergence KL gives. Let us consider, other two divergences:

- $L_1\left(p \| q\right) = \int_{\mathcal{X}} |p(x) - q(x)| \mathrm{d}x$
- $\chi^2\left(p \| q\right) = \int_{\mathcal{X}} \frac{|p(x) - q(x)|^2}{q(x)} \mathrm{d}x.$

Example 1.9 (KL versus L_1)**.**
Consider $p(x) = \mathrm{Uniform}\,(0, 1)$ and $q_n(x) = \mathbb{1}_{x \in [0, 1/n]} e^{-n} + \mathbb{1}_{x \in [1/n, 1]} c_n$, with $c_n \geq 0$ so that q integrates 1 ($n > 1$).

Let us first compute c_n explicitly. Since q_n must integrate to one, we require

$\int_0^{1/n} e^{-n}\,\mathrm{d}x + \int_{1/n}^1 c_n\,\mathrm{d}x = 1$, which yields

$$\frac{1}{n}e^{-n} + \left(1 - \frac{1}{n}\right) c_n = 1.$$

Solving for c_n, we obtain

$$c_n = \frac{1 - \frac{1}{n}e^{-n}}{1 - \frac{1}{n}} = \frac{n - e^{-n}}{n-1}. \tag{1.33}$$

Note that here, we have

$$L_1\left(p\|q_n\right) = \int_0^{1/n} |e^{-n} - 1|\mathrm{d}x + \int_{1/n}^1 |c_n - 1|\mathrm{d}x = \frac{|e^{-n} - 1|}{n} + \frac{n|c_n - 1|}{n-1} \tag{1.34}$$

$$\mathrm{KL}\left(p\|q_n\right) = \int_0^{1/n} -\log e^{-n}\mathrm{d}x + \int_{1/n}^1 -\log c_n \mathrm{d}x = 1 + \frac{-n}{n-1}\log c_n. \tag{1.35}$$

Now take $n \to \infty$. From eq. (1.33) we can see that c_n converges to 1. Therefore, $L_1\left(p\|q\right) \to 0$. However, note that $\mathrm{KL}\left(p\|q\right) \to 1$.

Example 1.10 (KL versus χ^2)**.**
Consider now $p_\epsilon = (1-\epsilon, \epsilon)$ and $q_\epsilon = (1-\epsilon^2, \epsilon^2)$ two Bernoulli distribution with different parameters. Again, we have:

$$\chi^2\left(p_\epsilon\|q_\epsilon\right) = \frac{||1 - \epsilon - 1 + \epsilon^2||^2}{1-\epsilon^2} + \frac{||\epsilon - \epsilon^2||^2}{\epsilon^2} = \frac{||\epsilon^2 - \epsilon||^2}{1-\epsilon^2} + ||1-\epsilon||^2 \tag{1.36}$$

$$\mathrm{KL}\left(p_\epsilon\|q_\epsilon\right) = (1-\epsilon)\log\frac{1-\epsilon}{1-\epsilon^2} + \epsilon\log\frac{\epsilon}{\epsilon^2} = (1-\epsilon)\log\frac{1}{1+\epsilon} + \epsilon\log\frac{1}{\epsilon}. \tag{1.37}$$

This time, taking $\epsilon \to 0$, we have $\mathrm{KL}\left(p_\epsilon\|q_\epsilon\right) \to 0$ (l'Hôpital's rule), but $\chi^2\left(p_\epsilon\|q_\epsilon\right) \to 1$

Remark 1.12.
The objective of these examples is to show that under different divergences, one can have different criteria of convergence. In the first case, q_n converges to p under L_1, but not under KL. In the second case, p_ϵ converges to q_ϵ under KL but not under ξ^2. This give a sense of *hierarchy* across divergences, where some are sais to induce stronger topologies than others. The stronger the topology, the more demanding the conditions for convergence (or fewer sequences are admitted to converge). In general, we consider KL as one of the stronger divergences (but there are some that are even stronger as we just saw).

Direct versus reverse KL. Since $\mathrm{KL}\left(p\|q\right)$ is not symmetric, we are interested in studying the *reverse* divergence $\mathrm{KL}\left(q\|p\right)$ and understanding how it relates its *direct* counterpart.

Since $P \ll Q$ is needed for $\mathrm{KL}\left(p\|q\right)$, it is required that $P \gg Q$ for $\mathrm{KL}\left(q\|p\right)$. This gives intuition of $\mathrm{KL}\left(p\|q\right)$ as a metric assessing how well q approximates p (and not viceversa); this is because if there is a set $A \subset \mathcal{X}$ such that $Q(A) = 0$ and $P(A) > 0$ is strongly penalised, unlike the opposite case.

Let us see a numerical example.

Example 1.11 (Assymetry of the KL between two Gaussians)**.**
The KL divergence between two Gaussians is

$$\mathrm{KL}\left(\mathcal{N}(\mu_0, \sigma_0) \| \mathcal{N}(\mu_1, \sigma_1)\right) = \log \frac{\sigma_1}{\sigma_0} + \frac{\sigma_0^2 + (\mu_0 - \mu_1)^2}{2\sigma_1^2} - \frac{1}{2}. \tag{1.38}$$

Let us consider $p = \mathcal{N}(0, 1)$ and $q = \mathcal{N}(0, v^2)$, and evaluate

- $\mathrm{KL}\left(p \| q\right) = \frac{1}{2}(\log v^2 + v^{-2} - 1)$
- $\mathrm{KL}\left(q \| p\right) = \frac{1}{2}(-\log v^2 + v^2 - 1)$.

Fig. 1.4 shows these functions, note how the penalisation strength depends on the direction.

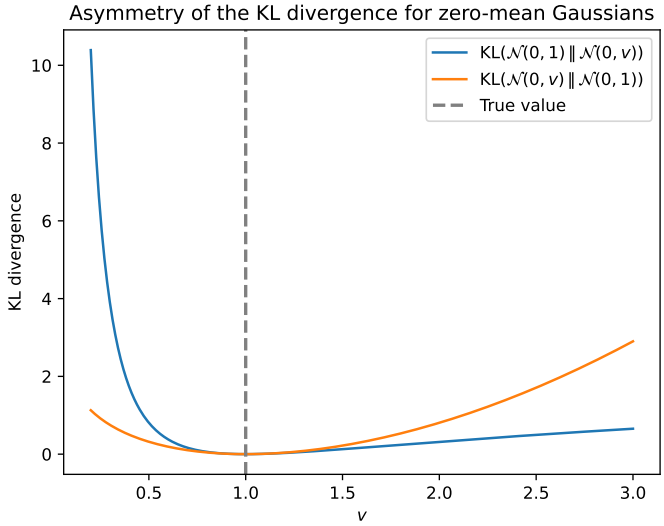

Figure 1.4: Direct and reverse KL for zero mean Gaussians as a function of the variance.

Example 1.12 (KL gradient flow)**.**
Let us now find the approximating q via optimisation for the above example. We can do this via optimisation. Differentiating eq. (1.38) wrt to μ_1 and σ_1, we have

$$\nabla_{\mu_1} \mathrm{KL}\left(\mathcal{N}(\mu_0, \sigma_0) \| \mathcal{N}(\mu_1, \sigma_1)\right) = \frac{(\mu_1 - \mu_0)}{\sigma_1^2} \tag{1.39}$$

$$\nabla_{\sigma_1} \mathrm{KL}\left(\mathcal{N}(\mu_0, \sigma_0) \| \mathcal{N}(\mu_1, \sigma_1)\right) = \frac{1}{\sigma_1} - \frac{\sigma_0^2}{\sigma_1^3} = \frac{\sigma_1^2 - \sigma_0^2}{\sigma_1^3} \tag{1.40}$$

Where it is clear the this is minimised for $q = p$. Additionally, we can build the gradient descent rule:

$$\mu_n \to \mu_n - \eta_\mu \frac{(\mu_n - \mu_0)}{\sigma_n^2} \tag{1.41}$$

$$\sigma_n \to \sigma_n - \eta_\sigma \frac{\sigma_n^2 - \sigma_0^2}{\sigma_n^3} \tag{1.42}$$

Figure 1.5 implements these recursions.

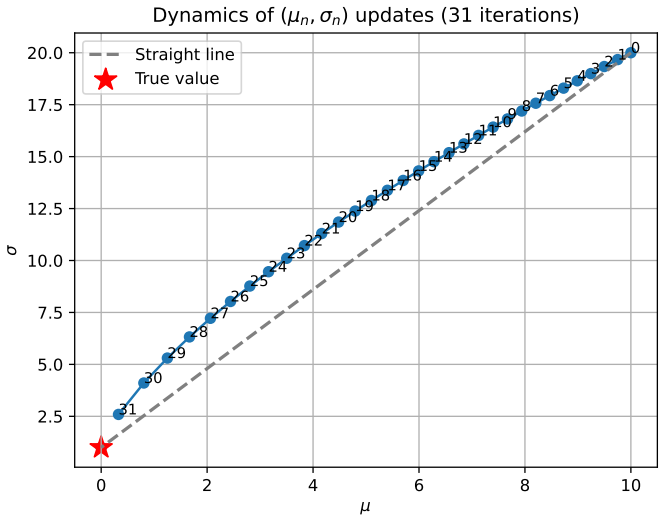

Figure 1.5: KL gradient flow between two Gaussians.

KL and maximum likelihood. Let us now return to the setting of the learning problem. Consider a true model given by p and iid observations $x_1, x_2, \ldots, x_N \sim p(x)$. We could use the KL to look for the best approximator of p with a given family of candidate models $\{q_\theta, \theta \in \Theta\}$. That is,

$$\theta^\star = \arg\min \mathrm{KL}\left(p \| q_\theta\right). \tag{1.43}$$

Though it sounds good, this is unfeasible in practice since p is unknown. However, note that there is a workaround to that. We can write the above expression as

$$\theta^\star = \arg\min \int_{\mathcal{X}} p(x) \log p(x) \mathrm{d}x - \int_{\mathcal{X}} p(x) \log q_\theta(x) \mathrm{d}x \tag{1.44}$$

$$= \arg\max \mathbb{E}_p[\log q_\theta(x)] \tag{1.45}$$

$$\approx \arg\max \sum_{x_i} \log q_\theta(x_i), \tag{1.46}$$

where the last approximation is due to Monte Carlo. This reveals that maximum likelihood is (asymptotically) equivalent to minimising the KL divergence between the model candidate and the (true) empirical distribution.

1.7 Concluding remarks

As we will throughout the module, when designing/chossing a PGM, we will be faced with the following scenarios.

- **Explicit-likelihood models:** These include classical statistical models such as Gaussians, exponentials, Bernoulli, Xi^2, log-normal, but also combinations or transformations that we construct as long as they have an explicit likelihood. These are Gaussian mixtures, piece-wise defined distributions, and amy pushforward model constructed through an invertible transformation so that its density can be calculated via the change of variable theorem. These last models are referred to as normalising flows, as they assume a Gaussian source measure.
- **Implicit models:** As the name suggests, these models are only implicitly defined via a data-generating mechanism, usually involving sampling. For instance, take a RV $Z \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$ and construct $X = T_\theta(Z)$, where T_θ is collection of neural networks which are sequentially applied to Z with the aim to replicate learnt dynamics that make X *flow* towards the desired distribution. Depending on the parametrisation, these models are known as score-based models, diffusion models, or flow matching.

1.7.1 Suggested exercises

1. **Generative vs discriminative modelling (theory).** Let (X, Y) be random variables with joint distribution $p(x, y)$.
   (a) Define what is meant by a *generative model* and a *discriminative model*.
   (b) Show how a generative model can be used to construct a classifier.
   (c) Discuss one advantage and one limitation of generative modelling relative to discriminative modelling.
2. **Information-theoretic objectives (theory).** Let p be a data-generating distribution and q_θ a parametric model.
   (a) Define the entropy $H(p)$, cross-entropy $H(p, q_\theta)$, and KL divergence $\mathrm{KL}(p\|q_\theta)$.
   (b) Show that maximising the log-likelihood of data sampled from p is equivalent to minimising $\mathrm{KL}(p\|q_\theta)$.
   (c) Explain why minimising $\mathrm{KL}(p\|q_\theta)$ and $\mathrm{KL}(q_\theta\|p)$ lead to qualitatively different approximations.
3. **Maximum likelihood density estimation (coursework).** You are given samples from a one-dimensional distribution.
   (a) Fit a Gaussian model by maximum likelihood.
   (b) Fit a mixture of Gaussians by maximum likelihood.
   (c) Empirically compare the learned models using log-likelihood and visual inspection.

4. **Forward and reverse KL divergence (coursework).** Consider approximating a multimodal target distribution using a unimodal Gaussian.

   (a) Numerically minimise $\mathrm{KL}(p\|q)$ and $\mathrm{KL}(q\|p)$.

   (b) Visualise the resulting solutions.

   (c) Explain the observed behaviour using the geometry of the KL divergence.

Week 2

Expectation Maximisation

NB: This is based on Chapter 9 of (Bishop, 2006).

2.1 Gaussian mixtures

Consider a dataset $\{x_1, x_2, \ldots, x : N\} \subset \mathbb{R}^d$. Our task is to partition this set into $K \in \mathbb{N}$ subsets; we will consider K known for now. Intuitively, each subset of points—referred to as a *cluster*—should share some common or similar patterns; a formal definition of similarity in this case will be ignored until needed.

A natural solution for this segmentation problem is to define K prototypes denoted $\{\mu_1, \mu_2, \ldots, \mu_K\} \subset \mathbb{R}^d$ and determine the assignment of each datapoint x_n to each prototype μ_k, according to a given criterion.

To solve this optimisation problem, we can define a set of binary variables $\{r_{nk}\}_{nk} \subset \{0, 1\}$, where

$$r_{nk} = 1 \iff x_n \text{ is assigned to } \mu_k. \tag{2.1}$$

Then, using the Euclidean distance as similarity criterion, the objective can be written as

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2. \tag{2.2}$$

The solution to the clustering problem obtained via the minimisation of the loss in eq. (2.2) is

$$r_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_j ||x_n - \mu_j||^2, \\ 0, & \text{if not.} \end{cases} \tag{2.3}$$

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}. \tag{2.4}$$

This solution can be calculated by iteratively implementing the above equations, which is known as the k-means algorithm.

Remark 2.1.

Observe that the k-means recursion ensures convergence in a finite number of steps:

this is because eq. (2.3) defines a discrete number of solutions, and (2.4) is the global optima for a given $\{r_{nk}\}_{nk}$.

There are some known drawbacks of k-means, for instance

- Speed: computing the assignment variables has a cost $\mathcal{O}(NK)$.
- It depends on the Euclidean distance that might not be robust to outliers
- It only provide hard assignments, not a degree of *responsibility*.

2.2 The Gaussian mixture model

Let us consider the following PGM:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k), \tag{2.5}$$

where $0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$, $\mu_k \in \mathbb{R}^d$ and $\Sigma_k \in \mathbb{R}^{d \times d}$.

This formulation seems to be an improved clustering model wrt K-means, since it—at least—allows for learning the shape (variance) of each cluster and admits the definition of a soft assignment variable.

However, note that the likelihood of this models is ill posed. Denoting the parameters by $\theta = \{\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}\}$ and the i.i.d. data $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, the log-likelihood is given by

$$l(\theta) = \log p(\mathbf{x}|\theta) = \log \prod_{n=1}^{N} p(x_n|\theta) = \sum_{n=1}^{N} \log p(x_n|\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k). \tag{2.6}$$

This objective can reach an infinite value if a Gaussian component is assigned to a single datapoint with a vanishing variance. Additionally, for each possible assignment, there are $K!$ different solutions that provide such assignment.

We will derive an equivalent formulation to the PGM above that admits a more interpretable and *stepwise* training procedure. To this end, let us introduce a set of K latent variables $\{z_k\} \subset \{0, 1\}$, $\sum_{k=1}^{K} z_k = 1$. We can write

$$p(x, z) = p(x|z)p(z). \tag{2.7}$$

Also, defining $p(z_k = 1) = \pi_k$, we can express the pmf/pdf:

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{2.8}$$

$$p(x|z) = \prod_{k=1}^{K} \mathcal{N}(\mu_k, \Sigma_k)^{z_k}, \tag{2.9}$$

with the marginal pdf over x as

$$p(x) = \sum_{k=1}^{K} p(z_k)p(x|z_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k). \tag{2.10}$$

Thus, showing that the formulations are equivalent.

In this formulation, let us define the *responsibilities* of the k-th component to explain the observation x given by

$$\gamma(z_k) \stackrel{\text{def}}{=} p(z_k = 1|x) = \frac{p(x|z_k = 1)p(z_k = 1)}{\sum_{j=1}^{K} p(x|z_j = 1)p(z_j = 1)} \tag{2.11}$$

$$= \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x; \mu_j, \Sigma_j)}. \tag{2.12}$$

Remark 2.2.
The latent-variable formulation of the GMM allows for direct sampling from that PGM: first sample $z \sim p(z) = \prod_{k=1}^{K}$, and then sample $x \sim p(x|z) = \prod_{k=1}^{K} \mathcal{N}(\mu_k, \Sigma_k)^{z_k}$. This is known as *ancestral sampling*.

2.3 Expectation Maximisation for GMMs

We will introduce a learning approach for PGMs that features a latent variable called Expectation Maximisation (EM). We will first present it in the particular case of the GMM model, and then in the general case.

The first order optimality conditions for the log-likelihood in eq. (2.6) give

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{2.13}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^\top \tag{2.14}$$

$$\pi_k = \frac{N_k}{N}, \tag{2.15}$$

where we have defined the effective number of samples per component as $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$.

Exercise 2.1.
Derive eqs. (2.13)-(2.15)

Remark 2.3.
Observe how the optimal mean and variance of each component is a weighted average of all the data points, where the weights are proportional to the responsibility (contribution) of that component to generation of the sample. Also, note that eqs. (2.13)-(2.15) can be considered as the soft-assignment version of the K-means solutions, with the additional flexibility of having an learnable expression for the shape of the clusters.

Eqs. (2.13)-(2.15) do not provide a direct closed-form solution, since they depend on the responsibilities $\gamma(z_{nk})$ which are functions of all the parameters. However, they can still be implemented in following the steps:

(E) Compute $\gamma(z_{nk}) = p(z_{nk} = 1|\mathbf{x})$.

(M) Use $\gamma(z_{nk})$ to compute eqs. (2.13)-(2.15).

2.4 An interpretation of EM

Let us now leave the GMM aside. In more general, perhaps abstract, terms, the goal of the EM algorithm is to find maximum likelihood solutions for latent variable models (LVMs) by breaking down the the optimisation problem into a functional approximation of the likelihood, and then the (simpler) optimisation of such approximation.

Recall our notation involving an observed variable x and a latent variable z. In general LVMs, the log-likelihood can be expressed as

$$\log p(x|\theta) = \int_{\mathcal{Z}} p(x|\theta, z)p(z|\theta)\mathrm{d}z. \tag{2.16}$$

NB: We treat both the discrete and continuous equivalently.

In general, direct optimisation of eq. (2.16) is difficult. Even calculating the above expression is only possible in limited cases, since mixtures do not mix well with the logarithm. In fact, even for likelihood in the exponential family, the mixture is not longer exponential and thus the application of the logarithm does not remove the exponential as in the single-Gaussian case.

Let us then consider the hypothetical scenario where we have access to the values of z alongside the observed x.

Definition 2.1.
We will refer to $\{\mathbf{x}, \mathbf{z}\}$ as the complete dataset, while $\mathbf{x}$ will be the *observed* or *incomplete* dataset.

The related likelihood to the complete dataset is

$$\log p(\mathbf{x}, \mathbf{z}|\theta) = \log \prod_{n=1}^{N} p(x_n, z_n|\theta) \tag{2.17}$$

$$= \log \prod_{n=1}^{N} p(x_n|z_n, \theta)p(z_n|\theta) \tag{2.18}$$

$$= \sum_{n=1}^{N} \log p(x_n|z_n, \theta) + \log p(z_n|\theta) \tag{2.19}$$

which we will assume is simpler to evaluate and optimise than $\log p(\mathbf{x}|\theta)$.

This, however, is impractical since $\mathbf{z}$ is unknown. An interesting interpretation of this optimisation problem is presented next.

Remark 2.4.

Since z_n is not observed, the complete-data log-likelihood in eq. (2.19) can be interpreted as a random function. Therefore, an alternative optimisation strategy is to estimate its expectation (E step) and then maximise the resulting deterministic expression (M step). At the end of the chapter, we will formally justify why taking the expectation is more than an intuition.

In more detail, this 2-step optimisation procedure results in moving from a candidate solution θ^{old} by first computing $p(\mathbf{z}|\mathbf{x}, \theta^{\text{old}})$ and then the expectation of the complete-data log-likelihood in eq. (2.19) given by

$$Q(\theta, \theta^{\text{old}}) = \sum_{z} \log p(\mathbf{x}, \mathbf{z}|\theta) p(\mathbf{z}|\mathbf{x}, \theta^{\text{old}}), \tag{2.20}$$

to finally reach an updated candidate solution θ^{new}

$$\theta^{\text{new}} = \arg\max_{\theta} Q(\theta, \theta^{\text{old}}). \tag{2.21}$$

Remark 2.5.

This procedure can also be used for maximum a posteriori estimation, in which case $Q(\theta, \theta^{\text{old}}) \to Q(\theta, \theta^{\text{old}}) + \log p(\theta)$ incorporates the prior over the parameter.

Now let us return to the GMM case and feed back these observations. For the GMM, the complete-data log-likelihood is

$$p(\mathbf{x}, \mathbf{z}|\theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}, \tag{2.22}$$

and thus the log-likelihood is

$$l(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \right), \tag{2.23}$$

which is tractable.

Remark 2.6.

The objective in eq. (2.23) is straightforward to optimise: since only one term in the k-sum is non-zero, the optimal mean and covariances can be computed in the same ways as the single Gaussian case. Furthermore, imposing the first order optimality condition and enforcing the $\sum_{k=1}^{K} \pi_k = 1$ via the Lagrangian, gives $\pi_k = \sum_{n=1}^{N} x_n / N$ directly.

Recall that this optima is a function of $\mathbf{z}$, and thus impossible to calculate directly, so we will calculate its expectation. To this end, we have

$$p(\mathbf{z}|\mathbf{x}, \theta) \propto p(\mathbf{z}, \mathbf{x}|\theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \underbrace{\pi_k^{z_{nk}} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}}_{\propto p(z_{nk}|x_n, \theta)}, \tag{2.24}$$

which means that $p(\mathbf{z}|\mathbf{x}, \theta)$ factorises wrt to n and k, and thus all the z_n are independent.

This is reasonable, since the cluster responsibilities over one sample should not affect the rest (due to the i.i.d. assumption).

The expectation of $\mathbf{z}$ can be computed as follows,

$$\begin{aligned}
\mathbb{E}\left(z_{nk}\right) &= 1 \cdot p(z_{nk} = 1|\mathbf{x}, \theta) + 0 \cdot p(z_{nk} = 0|\mathbf{x}, \theta) && (2.25)\\
&= p(z_{nk} = 1|\mathbf{x}, \theta) && (2.26)\\
&= \gamma(z_{nk}) && (2.27)\\
&\stackrel{\text{def}}{=} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}. && (2.28)
\end{aligned}$$

Note from eq. (2.23) that the objective is linear in z_{nk}, which makes the computation of its expectation straightforward:

$$\mathbf{E}_z \log p(\mathbf{x}, \mathbf{z}|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \left(\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)\right). \tag{2.29}$$

Exercise 2.2.
Show that GMM recovers K-means. For that, choose $p(x|\mu_k, \Sigma_k) = \mathcal{N}(x|\mu_k, \epsilon I)$ with $\epsilon > 0$ fixed to show that you recover a *soft-assignment* version of K-means. Then, take $\epsilon \to 0$ to recover vanilla K-means.

Exercise 2.3.
See the applications of EM to mixtures of Bernoulli and Bayesian linear regression in (Bishop, 2006).

Example 2.1.
Let us consider an implementation of the GMM training pipeline as described. Assuming $\mathbb{R}^2$ as the sample space, $K = 3$ clusters, $N = 400$ samples, Fig. 2.1 shows two choices for the initial condition. Then, after running the iterative training procedure, Fig. 2.2 shows the learnt clusters alongside the true values. Lastly, Fig. 2.3 shows the evolution of the likelihood per iteration. Note that both initialisations arrived at the same model, but the *good* initialisation was more efficient. The demo is available in the repository.

2.5 EM in its general form

Recall:

$$\underbrace{p(\mathbf{x}|\theta)}_{\text{difficult}} = \sum_{\mathbf{z}} \underbrace{p(\mathbf{x}, \mathbf{z}|\theta)}_{\text{easier}}. \tag{2.30}$$

We are interested in deriving EM as a model-approximation approach. To that end, let us consider a distribution over the latent variable $q(z)$; intuitively, this distribution will approximate $p(z|x, \theta)$. For any choice of q, the following holds:

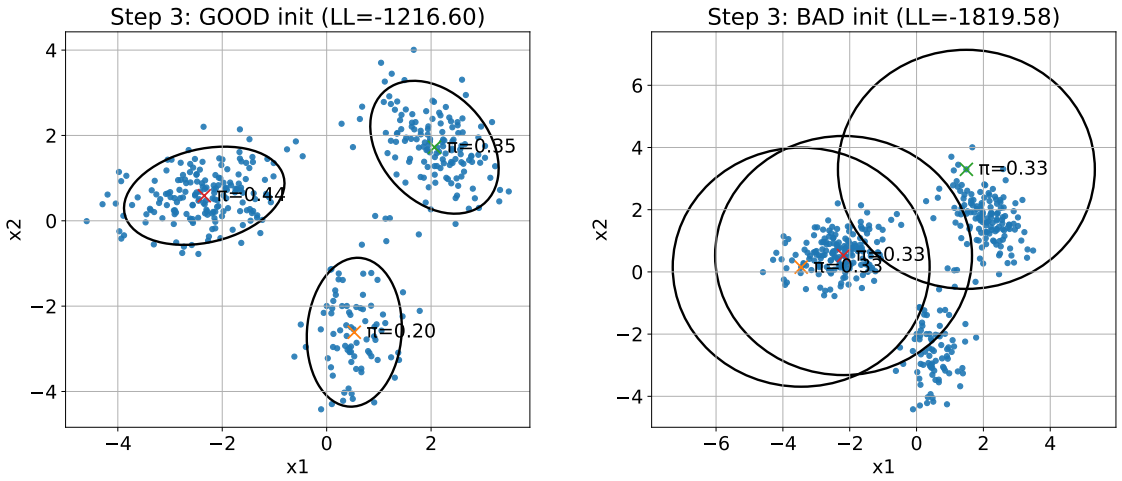

Figure 2.1: Two initialisations for the GMM model

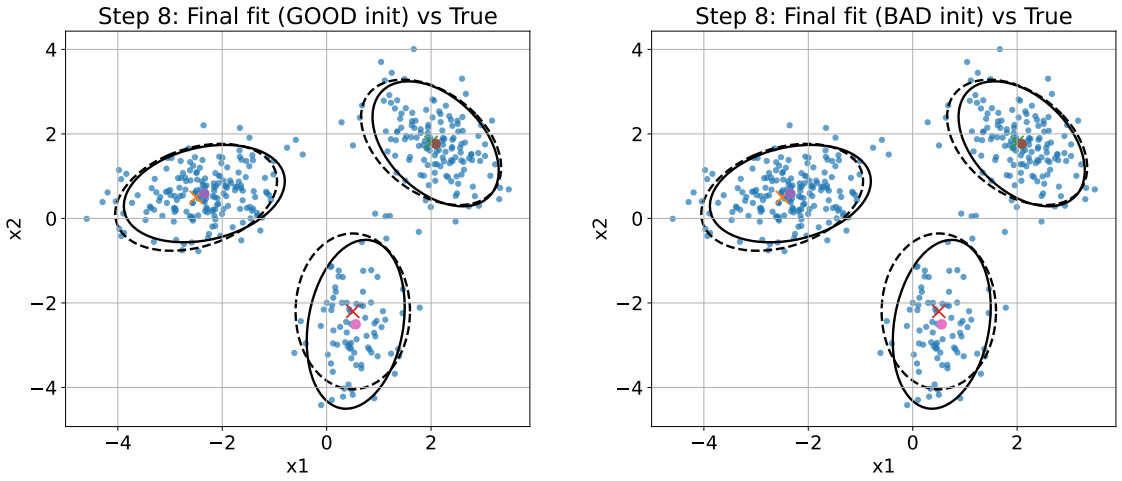

Figure 2.2: Solutions corresponding to the initialisations in Fig. 2.1.

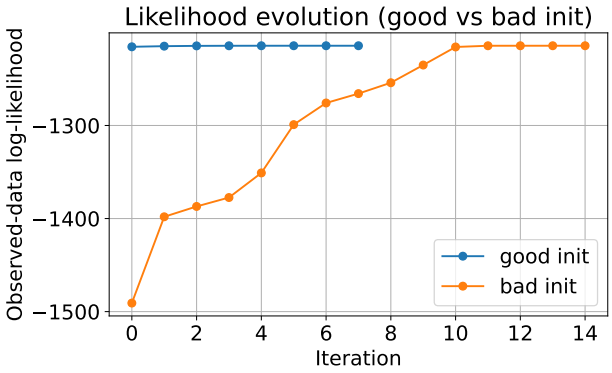

Figure 2.3: Evolution of the log-likelihoodfor both initialisations in Fig. 2.1.

$$
\begin{aligned}
\log p(x|\theta) &= \sum_z q(z) \log p(x|\theta) && \leftarrow \quad \log p(x|\theta) \text{is constant wrt } z \\
&= \sum_z q(z) \log \left(p(x|\theta) \frac{p(z|x,\theta) q(z)}{p(z|x,\theta) q(z)} \right) && \leftarrow \quad \text{multiply by 1} \\
&= \sum_z q(z) \log \left(\frac{p(x,z|\theta)}{q(z)} \cdot \frac{q(z)}{p(z|x,\theta)} \right) && \leftarrow \quad \text{arrange} \\
&= \underbrace{\sum_z q(z) \log \left(\frac{p(x,z|\theta)}{q(z)} \right)}_{\mathcal{L}(q,\theta)} + \underbrace{\sum_z q(z) \log \left(\frac{q(z)}{p(z|x,\theta)} \right)}_{\mathrm{KL}(q(z) \| p(z|x,\theta))}. && \leftarrow \quad \text{split}
\end{aligned}
$$

Remark 2.7.
Recall that the KL is always non-negative, meaning that $\mathcal{L}(q,\theta)$ is a lower bound for $\log p(x|\theta)$.

EM breaks the ML problem into two simpler problems related to the computation of a lower bound and its optimisation. The objective above can be minimised in two stages:

- First, the distribution q is chosen in order to minimise the term $\mathrm{KL}\left(q(z) \| p(z|x,\theta)\right)$, where, the optimal solution in $q(z) = p(z|x,\theta)$.
- Then, using that choice for q, the term $\mathcal{L}(q,\theta)$ is optimised. Notice that this term is an expectation wrt q.

Remark 2.8.
This view formalises the intuition that we presented for the GMM case. The choice of the expectation of the complete log-likelihood is not arbitrary, but follows from finding the optimal approximating distribution in the KL sense.

Monotonicity of EM. Recall the variational decomposition (valid for any distribution q on z)

$$\log p(x|\theta) = \mathcal{L}(q,\theta) + \mathrm{KL}\left(q(z) \| p(z|x,\theta)\right), \tag{2.31}$$

where

$$\mathcal{L}(q,\theta) := \sum_z q(z) \log \frac{p(x,z|\theta)}{q(z)}. \tag{2.32}$$

Since $\mathrm{KL}\left(\cdot\|\cdot\right) \geq 0$, we have $\log p(x|\theta) \geq \mathcal{L}(q,\theta)$ for all (q,θ).

Let θ^t be the current iterate. The E-step sets

$$q^{t+1}(z) := p(z|x,\theta^t), \tag{2.33}$$

for which $\mathrm{KL}\left(q^{t+1}(z) \| p(z \mid x,\theta^t)\right) = 0$. Plugging into (2.31) yields

$$\log p(x|\theta^t) = \mathcal{L}(q^{t+1},\theta^t). \tag{2.34}$$

The M-step then chooses

$$\theta^{t+1} \in \arg\max_{\theta} \ \mathcal{L}(q^{t+1}, \theta), \tag{2.35}$$

hence

$$\mathcal{L}(q^{t+1}, \theta^{t+1}) \geq \mathcal{L}(q^{t+1}, \theta^t) = \log p(x|\theta^t). \tag{2.36}$$

Finally, since $\log p(x|\theta) \geq \mathcal{L}(q^{t+1}, \theta)$ for any θ, we obtain

$$\log p(x|\theta^{t+1}) \ \geq \ \mathcal{L}(q^{t+1}, \theta^{t+1}) \ \geq \ \log p(x|\theta^t). \tag{2.37}$$

Therefore EM produces a non-decreasing sequence of log-likelihood values.

Remark 2.9 (Generalised EM)**.**
The monotonicity argument above only requires that the M-step returns θ^{t+1} such that $\mathcal{L}(q^{t+1}, \theta^{t+1}) \geq \mathcal{L}(q^{t+1}, \theta^t)$. Thus, it is not necessary to fid the exact (local) optimum of $\mathcal{L}$ in the M-step; any update that increases $\mathcal{L}$ yields a *generalised EM* (GEM) procedure with the same monotonicity guarantee. This observation will be key for the variational inference part.

Does EM fix ill-posed maximum likelihood? For several latent-variable models, the maximum likelihood (ML) objective is *ill-posed*, since the log-likelihood can be unbounded. A canonical example is the Gaussian mixture model (GMM): as we saw in class, for any datapoint $x_n \in \mathbb{R}^d$ and any $\epsilon > 0$, consider a mixture component k with

$$\mu_k = x_n, \qquad \Sigma_k = \epsilon I.$$

Then

$$\mathcal{N}(x_n; \mu_k, \Sigma_k) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2} = (2\pi)^{-d/2}\epsilon^{-d/2},$$

so the corresponding contribution to $\log p(x_n|\theta)$ diverges as $\epsilon \to 0$. Consequently, $\sup_\theta \log p(x|\theta) = +\infty$ for unconstrained GMM maximum likelihood.

EM *does not* resolve this ill-posedness, because it is still (attempting to) maximise the same ML objective; it only changes the optimisation route. In fact, the EM updates can actively move toward singular solutions: if a component collapses around a datapoint, the E-step tends to assign it responsibility close to 1 for that datapoint, and the M-step covariance update (a responsibility-weighted empirical covariance) can shrink toward a rank-deficient matrix unless additional constraints are imposed.

Remark 2.10 (How to avoid degeneracy)**.**
To obtain well-posed estimation one typically modifies the objective or the parameter space, e.g.

- **MAP (penalised) EM:** maximise $\log p(x \mid \theta) + \log p(\theta)$ by placing a prior on parameters (e.g. inverse-Wishart prior on Σ_k, Dirichlet prior on mixing weights).
- **Constrained ML:** enforce $\Sigma_k \succeq \sigma^2_{\min} I$ or tie covariances (e.g. $\Sigma_k = \Sigma$).
- **Regularisation / damping:** add a penalty to discourage small determinants, or replace $\Sigma_k \leftarrow \Sigma_k + \epsilon I$ after updates.
- **Alternative components:** heavier-tailed mixtures (e.g. Student-t mixtures) can

reduce single-point capture.

In short: EM guarantees non-decreasing likelihood; when the likelihood is *unbounded above*, monotone ascent is still compatible with divergence toward a singular solution.

Remark 2.11.
EM does not change the (possibly ill-posed) ML objective; it only changes the optimisation procedure. In particular, it replaces direct maximisation of $\log p(x|\theta)$ by alternating easier subproblems (posterior inference and complete-data fitting) and guarantees monotone progress, which often stabilises learning in practice even though degeneracies may still exist in principle.

2.6 Concluding remarks

- **EM as coordinate ascent on a lower bound.** EM alternates between (i) an *inference* step, selecting $q(z) = p(z|x, \theta)$ to tighten a variational lower bound, and (ii) a *learning* step, updating θ to increase that bound:

$$q^{t+1}(z) = p(z|x, \theta^t), \qquad \theta^{t+1} \in \arg\max_{\theta} \mathbf{E}_{q^{t+1}}[\log p(x, z|\theta)].$$

- **Inference vs. learning.** Latent-variable models are expressive, but at the cost of posterior inference. EM addresses this inference need, where the E-step performs posterior computation and the M-step performs parameter fitting.

- **Feasibility of EM.** EM is particularly effective when the complete-data likelihood belongs to a tractable family (often exponential-family), leading to closed-form M-steps. When the exact posterior is intractable, one can replace the E-step by a restricted family q (variational EM).

- **Limitations.** EM is monotone but generally converges to a stationary point, so performance depends on initialization. Moreover, EM does not fix ill-posed ML objectives (e.g. degeneracy in GMMs) without additional constraints or priors.

These ideas generalise beyond mixtures: many modern learning procedures can be understood as optimising tractable surrogates (bounds) of intractable objectives, with EM providing a canonical template.

2.6.1 Suggested exercises

1. **Latent variable models and incomplete data (theory).**

   (a) Define a latent variable model and distinguish between complete and incomplete data.

   (b) Explain why direct maximisation of the marginal likelihood is often intractable.

   (c) Describe how the introduction of latent variables simplifies modelling but complicates inference.

2. **Expectation–Maximisation algorithm (theory).** Consider a latent variable model with parameters θ.
   (a) Derive the EM algorithm starting from the marginal log-likelihood.
   (b) Define the Q-function and explain the role of the E-step and the M-step.
   (c) Prove that each EM iteration does not decrease the log-likelihood.
3. **Gaussian mixture models (coursework).**
   (a) Implement the EM algorithm for Gaussian mixture models.
   (b) Investigate the effect of initialisation on convergence.
   (c) Illustrate the relationship between k-means and GMMs by varying the covariance structure.
4. **Likelihood degeneracy and regularisation (coursework).**
   (a) Demonstrate empirically the likelihood degeneracy of Gaussian mixture models.
   (b) Propose and implement at least one regularisation strategy.
   (c) Analyse how regularisation affects the learned parameters and likelihood.

Week 3

Approximate Inference

NB: This is based on (Andrieu, de Freitas, Doucet & Jordan, 2003), (Blei, Kucukelbir & McAuliffe, 2017), and Chapter 10 of (Bishop, 2006).

3.1 Motivation: intractable posteriors

Bayesian inference in generative models. A probabilistic generative model specifies a joint distribution over observed variables $x \in \mathcal{X}$ and latent variables $z \in \mathcal{Z}$:

$$p_\theta(x, z) = p_\theta(z)\, p_\theta(x \mid z),$$

where θ denotes model parameters.

Given observations x, Bayesian inference aims to compute the posterior distribution

$$p_\theta(z \mid x) = \frac{p_\theta(x, z)}{p_\theta(x)}, \qquad p_\theta(x) = \int p_\theta(x, z)\, dz.$$

The posterior encodes all uncertainty about the latent structure z after observing the data, and is the central object of interest in Bayesian modeling.

Remark 3.1.

From now on, we will drop the explicit dependency of the parameter. From a Bayesian standpoint, we will consider that the latent variable z encapsulates all (random) unknowns. This includes global variables such as parameters, and local variables such as cluster assignments.

Example 3.1 (Hierarchical Gaussian Mixture Model)**.**

To illustrate the role of both global and local latent variables in a PGM, we consider the following hierarchical GMM.

Let $x_n \in \mathbb{R}^d$, $n = 1, \ldots, N$, denote the observed data. The latent variables are:

- mixture weights $\pi = (\pi_1, \ldots, \pi_K)$,
- cluster means $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$,
- cluster assignments $z_n \in \{1, \ldots, K\}$.

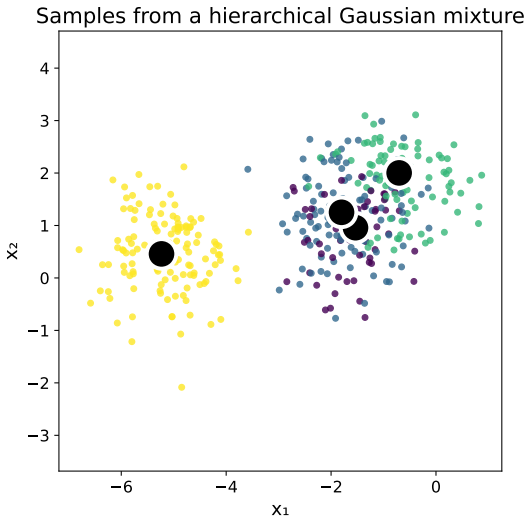

Figure 3.1: Samples from a hierarchical GMM: 2 dimensions, $N = 400$ samples.

The generative process is:

$$\begin{aligned} \pi &\sim \text{Dirichlet}(\alpha), \\ \mu_k &\sim \mathcal{N}(m_0, \Sigma_0), \qquad k = 1, \ldots, K, \\ z_n \mid \pi &\sim \text{Categorical}(\pi), \qquad n = 1, \ldots, N, \\ x_n \mid z_n, \{\mu_k\} &\sim \mathcal{N}(\mu_{z_n}, \Sigma). \end{aligned}$$

The resulting joint distribution factorizes as

$$p(x, z, \mu, \pi) = p(\pi) \prod_{k=1}^{K} p(\mu_k) \prod_{n=1}^{N} p(z_n \mid \pi)\, p(x_n \mid z_n, \mu).$$

Given observations $x_{1:N}$, the inference task is to approximate the posterior

$$p(z_{1:N}, \mu_{1:K}, \pi \mid x_{1:N}),$$

which is analytically intractable due to the hierarchical coupling between the latent variables. In the same spirit, the marginal given by

$$p(x_{1:N}) = \int p(z_{1:N}, \mu_{1:K}, \pi, x_{1:N}) \mathrm{d}z_{1:N} \mathrm{d}\mu_{1:K} \mathrm{d}\pi$$

is also analytically intractable. Fig. 3.1 shows an implementation of this model.

Example 3.2 (Bayesian Linear Regression)**.**
Let $\{(x_n, y_n)\}_{n=1}^{N}$ be observed data, where $x_n \in \mathbb{R}^d$ are inputs and $y_n \in \mathbb{R}$ are outputs. The latent variable is the regression weight vector $w \in \mathbb{R}^d$, which is the parameter of the model. Notice that in this case the parameter is the only global latent variable and there are no local latent variables.

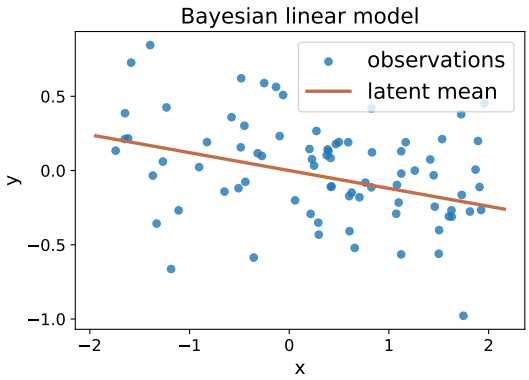

Figure 3.2: Samples from a Bayesian linear model: 1 dimensions, $N = 400$ samples.

The generative model is defined by

$$w \sim \mathcal{N}(m_0, \Sigma_0), \qquad y_n \mid w, x_n \sim \mathcal{N}(x_n^\top w, \sigma^2), \quad n = 1, \ldots, N.$$

Denoting the input matrix $X \in \mathbb{R}^{N \times d}$ with rows $x_n^\top$, and the observation vector $y = (y_1, \ldots, y_N)^\top$, the likelihood can be written compactly as

$$y \mid w, X \sim \mathcal{N}(Xw, \sigma^2 I_N).$$

The joint distribution factorises as

$$p(y, w \mid X) = p(w) \prod_{n=1}^{N} p(y_n \mid w, x_n).$$

Given observations (X, y), the posterior distribution over the latent weights is

$$p(w \mid X, y) \propto p(w) \prod_{n=1}^{N} p(y_n \mid w, x_n),$$

which admits the closed-form Gaussian solution

$$p(w \mid X, y) = \mathcal{N}(m_N, \Sigma_N),$$

with

$$\Sigma_N^{-1} = \Sigma_0^{-1} + \frac{1}{\sigma^2} X^\top X, \quad m_N = \Sigma_N \left(\Sigma_0^{-1} m_0 + \frac{1}{\sigma^2} X^\top y \right).$$

This model provides a baseline where exact Bayesian inference is tractable. Fig. 3.2 shows samples from this model alongside the linear function corresponding to the latent weight.

Example 3.3 (State space model)**.**
Let us now consider a latent variable model describing the evolution of a dynamical system observed through noisy measurements.

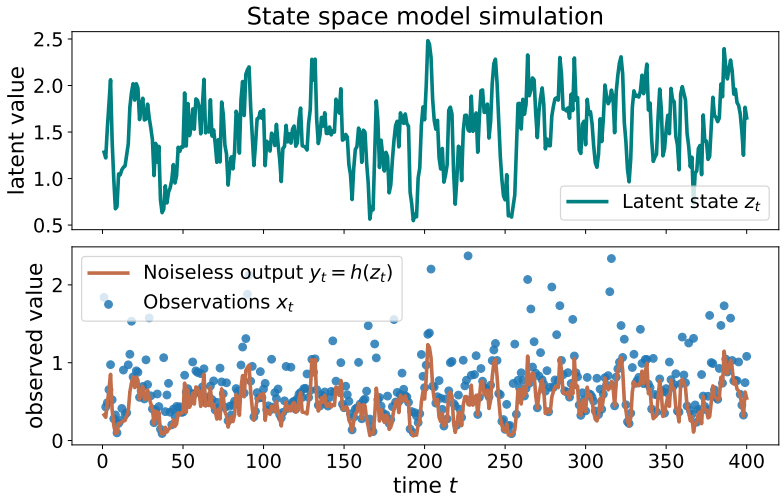

Figure 3.3: Samples from a nonlinear state space model: 1 dimension, $N = 400$ samples.

Let $z_t \in \mathbb{R}^m$, $t = 1, \ldots, T$, denote the latent state at time t, and let $x_t \in \mathbb{R}^d$ denote the corresponding observation. The latent states and observations are linked through a transition model and an observation model.

The generative process is defined as follows:

$$z_1 \sim p(z_1), \tag{3.1}$$

$$z_t = f(z_{t-1}) + \epsilon_t, \qquad t = 2, \ldots, T, \tag{3.2}$$

$$x_t = h(z_t) + \eta_t, \qquad t = 1, \ldots, T. \tag{3.3}$$

where $f : \mathbb{R}^m \to \mathbb{R}^m$ is the (possibly nonlinear) state transition function, $h : \mathbb{R}^m \to \mathbb{R}^d$ is the observation function, and $\epsilon_t \sim p_{\text{transition}}$ and $\eta_t \sim p_{\text{observation}}$ are the process and observation noise sources respectively.

The joint distribution over latent states and observations factorizes as

$$p(x_{1:T}, z_{1:T}) = p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(x_t \mid z_t).$$

Given observations $x_{1:T}$, the inference task is to approximate the posterior distribution

$$p(z_{1:T} \mid x_{1:T}),$$

which is analytically tractable only in the linear–Gaussian case (f and h linear), and otherwise requires approximate inference methods. Fig. 3.3 shows a sample form this model.

Approximate inference replaces the true posterior $p(z|x)$ with an approximation, say $q(z|x)$ that is as close a possible to the true posterior, while at the same time allows for computing expectations and scalable computation. In this sense, there are two dominant paradigms for computing q.

- Monte Carlo methods, which approximate the posterior via samples $z^{(s)} \sim p(z \mid x)$. This estimate is asymptotically exact but often computationally expensive.
- Variational methods, which provides a functional approximation of the posterior by solving an optimization problem:

$$q^\star(z) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\Big(q(z) \,\|\, p(z \mid x)\Big),$$

where $\mathcal{Q}$ is a tractable family of distributions. Variational approximations are fast and scalable, but introduce bias.

3.2 Markov chain Monte Carlo

Recall that Bayesian inference reduces to computing expectations of the form

$$\mathbb{E}_{p(z|x)}[f(z)] = \int f(z)\, p(z \mid x)\, dz.$$

When direct evaluation of this integral is infeasible, Monte Carlo methods approximate it using random samples:

$$\mathbb{E}_{p(z|x)}[f(z)] \;\approx\; \frac{1}{S}\sum_{s=1}^{S} f(z^{(s)}), \qquad z^{(s)} \sim p(z \mid x).$$

The law of large numbers guarantees that this estimator converges almost surely as $S \to \infty$. Furthermore, the root-mean-square error of this estimator decreases at a rate $1/\sqrt{S}$

However, the challenge in Bayesian inference is to obtains the samples $\{z^{(s)}\}_s$. Recall that the posterior takes the form

$$p(z \mid x) = \frac{p(x, z)}{p(x)},$$

Although the joint density $p(x, z)$ is often available, the normalizing constant

$$p(x) = \int p(x, z)\, dz = \int p(x \mid z) p(z)\, dz.$$

has an integral form that is rarely tractable and expensive to compute numerically in high dimensions. typically intractable. As a consequence, direct sampling from $p(z \mid x)$ is not possible.

The rational behind Markov Chain Monte Carlo (MCMC) methods is to construct a Markov chain whose limiting stationary distribution is the desired posterior.

Recall that a Markov chain $\{z^{(t)}\}_{t \geq 0}$ is defined by a transition kernel $T(z' \mid z)$, where

$$\mathbb{P}(z^{(t+1)} = z' \mid z^{(t)} = z) = T(z' \mid z).$$

For this Markov chain to have the posterior $p(z \mid x)$ as its limiting distribution, two conditions need to hold. First, the posterior has to be the chain's stationary distribution, that is,

$$\int p(z \mid x)\, T(z' \mid z)\, dz = p(z' \mid x).$$

This means that if the chain starts at $z \sim p(z \mid x)$, then after one step the chain remains in the same distribution. Second, the chain should converge to its stationary distribution.

A sufficient (but not necessary) condition for stationarity is the *detailed balance* condition,

$$p(z \mid x)\, T(z' \mid z) = p(z' \mid x)\, T(z \mid z'),$$

which implies that $p(z \mid x)$ is a stationary distribution of the Markov chain. The detailed balance condition states that, under the stationary distribution, the probability flow from state z to state z' is exactly balanced by the flow from z' to z. As a result, there is no net movement of probability mass, and the target distribution remains invariant under the Markov chain dynamics.

To ensure convergence to the stationary distribution from an arbitrary initial state, additional regularity conditions are required. In particular, the Markov chain should be *irreducible*, meaning that it is possible to reach any state from any other state with positive probability, and *aperiodic*, meaning that the chain does not get trapped in deterministic cycles.

Under these conditions, the distribution of $z^{(t)}$ converges to $p(z \mid x)$ as $t \to \infty$, regardless of the initial state.

Metropolis Hastings. A direct construction of the transition kernel that fulfills the convergence conditions above can be achieved in two steps. First, by sampling from an arbitrary proposal distribution $q(z' \mid z)$, and then correcting the sample for the discrepancy between q and the target distribution $p(z \mid x)$.

Given the current state z, the Metropolis–Hastings update proceeds as follows:

1. Propose a new state
$$z' \sim q(z' \mid z).$$

2. Accept the proposal with probability
$$\alpha(z, z') = \min\left(1,\ \frac{p(z' \mid x)\, q(z \mid z')}{p(z \mid x)\, q(z' \mid z)}\right). \tag{3.4}$$

   Meaning that the new sample is

$$z^{(t+1)} = \begin{cases} z', & \text{with probability } \alpha(z, z'), \\ z, & \text{otherwise.} \end{cases}$$

Remark 3.2.

MH does not require knowledge of the normalising constant $p(x)$. As a consequence,

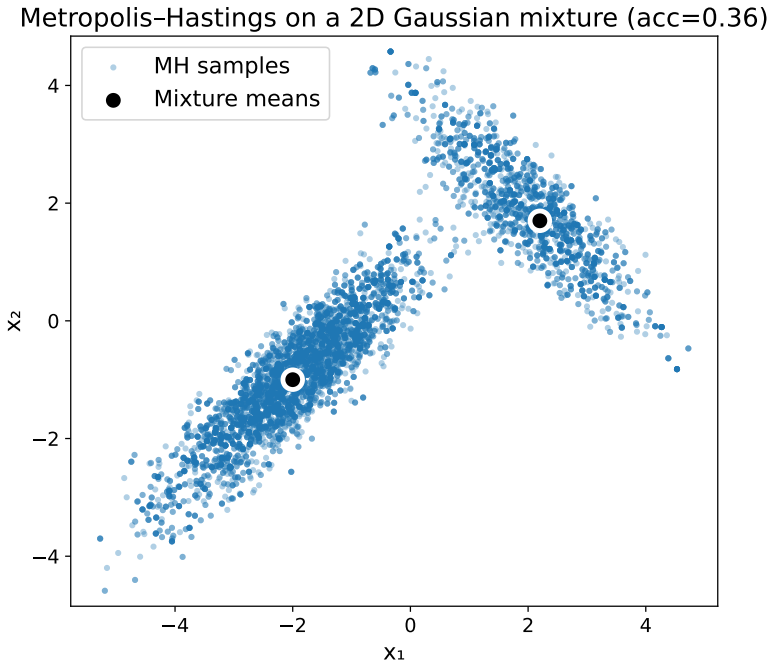

Figure 3.4: Samples from a 2D 2-Gaussian mixture using Metroplois Hastings: $N = 10,000$ samples.

since $p(z \mid x) \propto p(x, z)$, the acceptance probability can be computed using the joint density $p(x, z)$.

Remark 3.3.

The Metropolis–Hastings transition kernel satisfies the detailed balance condition:

$$p(z \mid x)\, T(z' \mid z) = p(z' \mid x)\, T(z \mid z').$$

This implies that $p(z \mid x)$ is a stationary limiting distribution of the chain.

Remark 3.4.

A particular instance of the MH algorithm can be identified by choosing a symmetric proposal, that is $q(z \mid z') = q(z' \mid z)$. In this case, the acceptance probability reduces to

$$\alpha(z, z') = \min\left(1,\ \frac{p(z' \mid x)}{p(z \mid x)}\right).$$

This is know as the Metropolis method.

A common choice for the proposal in practice is simply a random walk

$$q(z' \mid z) = \mathcal{N}(z, \sigma^2 I).$$

However, while simple to implement,this random-walk MH can mix poorly in high-dimensional or highly correlated posteriors.

Fig. 3.5 shows an implementation of Metropolis Hastings to sample from a mixture of 2 Gaussians using a single Gaussian as proposal.

Gibbs sampling. To develop a MH method that exploits conditional structure in the posterior distribution, the proposal over each coordinate of z can be coupled to previously sampled coordinates. To this end, denote

$$z = (z_1, \dots, z_d),$$

and update each component by sampling from its conditional distribution:

$$z_i \sim p(z_i \mid z_{-i}, x),$$

where z_{-i} denotes all components except z_i.

A complete Gibbs update consists of sequentially sampling:

$$\begin{aligned} z_1 &\sim p(z_1 \mid z_2, \dots, z_d, x), \\ z_2 &\sim p(z_2 \mid z_1, z_3, \dots, z_d, x), \\ &\vdots \\ z_d &\sim p(z_d \mid z_1, \dots, z_{d-1}, x). \end{aligned}$$

To see that Gibbs is a particular instance of MH, let us first denote the target posterior by $\pi(z) = p(z \mid x)$, and consider a fixed index $i \in \{1, \dots, d\}$. To update the i-th coordinate while keeping z_{-i} fixed, Gibbs samples directly from the conditional posterior

$$z_i' \sim \pi(z_i \mid z_{-i}),$$

and defines the proposed sample

$$z' = (z_i', z_{-i}).$$

For the complete variable z, this corresponds to a proposal distribution

$$\begin{aligned} q(z' \mid z) &= q(z_i', z_{-i}' \mid z_i, z_{-i}) \\ &= q(z_i' \mid z_i, z_{-i}, z_{-i}') q(z_{-i}' \mid z_i, z_{-i}) \\ &= q(z_i' \mid z_{-i}) q(z_{-i}' \mid z_{-i}) \\ &= \pi(z_i' \mid z_{-i})\, \mathbf{1}\{z_{-i}' = z_{-i}\}. \end{aligned} \tag{3.5}$$

To compute the acceptance probability in eq. (3.4), we can factorise

Using the factorization

$$\pi(z) = \pi(z_{-i})\, \pi(z_i \mid z_{-i}), \qquad \pi(z') = \pi(z_{-i})\, \pi(z_i' \mid z_{-i}),$$

together with the proposal in eq. (3.5) (recall that $z_{-i}' = z_{-i}$)

$$q(z' \mid z) = \pi(z_i' \mid z_{-i}), \qquad q(z \mid z') = \pi(z_i \mid z_{-i}),$$

we obtain

$$\frac{\pi(z')\, q(z \mid z')}{\pi(z)\, q(z' \mid z)} = \frac{\pi(z_{-i})\, \pi(z_i' \mid z_{-i})\, \pi(z_i \mid z_{-i})}{\pi(z_{-i})\, \pi(z_i \mid z_{-i})\, \pi(z_i' \mid z_{-i})} = 1.$$

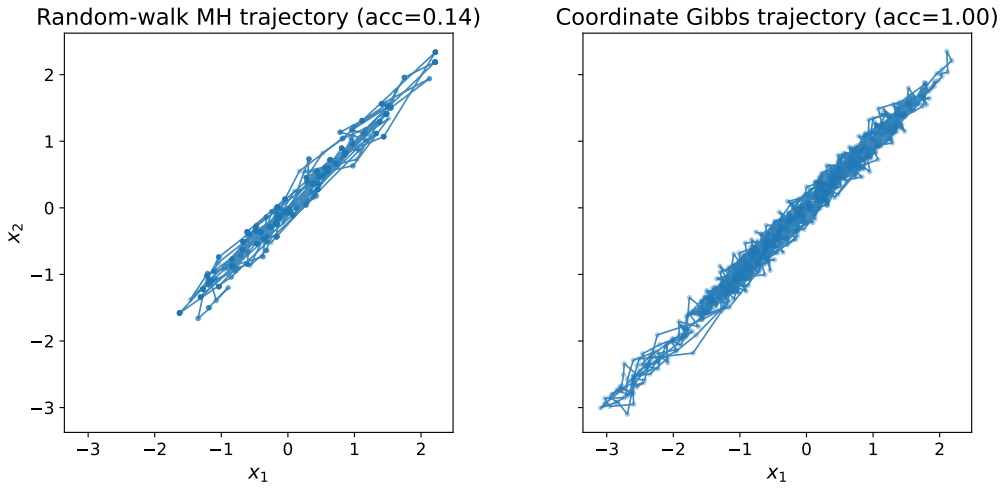

Figure 3.5: Samples from a 2D highly correlated Gaussian using Gibbs (right) and ;Metropolis Hastings (right): $N = 1,000$ samples.

Therefore, the acceptance probability is

$$\alpha(z, z') = 1,$$

meaning that the proposed move is always accepted.

Remark 3.5.

Gibbs sampling is a special case of the Metropolis–Hastings algorithm in which the proposal distribution is the exact full conditional distribution, and the acceptance probability is identically equal to one. Furthermore, each update (not necessarily the full update) leaves the joint posterior invariant.

Practical considerations. There are a number of implementation practices to be taken into account when suing MCMC.

Burn-in period: Though MH is guaranteed to converge to its stationary distribution, the fact that the chain starts from an arbitrary distribution implies that the initial samples will not be representative of the target posterior. Therefore, the initial sampels should be discarded as invalid samples from $p(z \mid x)$. When the samples are representative from the target, we say that the chain has *mixed*. Furthermore, assessment of this convergence must be assessed empirically.

Autocorrelation: The samples used for Monte Carlo integration should be i.i.d. samples from the target (posterior) distributions; in fact, when correlated samples are used it is the *effective sample size* that governs the quality of the approximation. Since the samples are generated by a chain with a correlated transition kernel, consecutive samples are not independent (in general correlated) by construction. To alleviate this, after sampling from the chain we should *thin* the chain, that is, to consider a subset of samples, for instance, one every 50 samples.

Fig. 3.5 shows an implementation of Gibbs and Metropolis Hastings to sample from a correlated Gaussian. Observe how with 1,000 samples Gibbs fully explores the support

of the distribution while MH struggle to propose valid sample.

3.3 Variational inference

The evidence lower bound (ELBO). Rather than trying to compute the posterior $p(z \mid x)$ directly, we will specify a family of densities over z termed $\mathcal{Q}$, where each $q(\cdot) \in \mathcal{Q}$ is a candidate approximation to the posterior. Our goal will ve then to find the best candidate within $\mathcal{Q}$. Our optimality criterion will be the will be the minimisation of the KL divergence, that is, we will find

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q(z) \| p(z \mid x)\right). \tag{3.6}$$

Remark 3.6.
The more general or comprehensive the family $\mathcal{Q}$, the more difficult it is to find the optimal $q^*(z)$.

Observe that solving eq. (3.6) is unfeasible, since it requires $p(z \mid x)$ or equivalently $p(x)$. However, notice that since

$$\mathrm{KL}\left(q(z) \| p(z \mid x)\right) = \int q(z) \log q(z) \mathrm{d}z - \int q(z) \log p(z, x) \mathrm{d}z + \underbrace{\int q(z) \log p(x) \mathrm{d}z}_{\log p(x)}$$

the dependence of the objective on $\log p(x)$ is via a constant which can be ignored when optimising wrt $q(z)$. Therefore, the optimisation problem in eq. (3.6) is equivalent to maximising the evidence lower bound (ELBO) given by

$$\mathrm{ELBO} \stackrel{\mathrm{def}}{=} \int q(z) \log p(z, x) \mathrm{d}z - \int q(z) \log q(z) \mathrm{d}z. \tag{3.7}$$

Let us observe the following decomposition of the ELBO:

$$\begin{aligned} \mathrm{ELBO} &= \int q(z) \log p(z, x) \mathrm{d}z - \int q(z) \log q(z) \mathrm{d}z \\ &= \int q(z) \log p(x \mid z) \mathrm{d}z + \int q(z) \log p(z) \mathrm{d}z - \int q(z) \log q(z) \mathrm{d}z \\ &= \mathbb{E}\left(\log p(x \mid z)\right) - \mathrm{KL}\left(q(z) \| p(z)\right), \end{aligned} \tag{3.8}$$

where all the expectations in this section will be wrt $q(z)$.

Maximising the ELBO in eq. (3.8) is a balance between two terms. The first one seeks to assign the mass of q to the values of z that explain the observations x, while the second one ensures that $q(z)$ is close to the prior $p(z)$.

Remark 3.7.
Maximising the ELBO recovers the usual likelihood / prior trade off.

To justify the name of this objective, let us see how it relates to the so called *evidence*

$\log p(x)$. Using the expression in eq. (3.7), we have

$$\begin{aligned}\log p(x) - \text{ELBO} &= \int q(z) \log p(x) \mathrm{d}z - \int q(z) \log p(z, x) \mathrm{d}z + \int q(z) \log q(z) \mathrm{d}z \\ &= \int q(z) \log \frac{1}{p(z \mid x)} \mathrm{d}z + \int q(z) \log q(z) \mathrm{d}z \\ &= \mathrm{KL}\left(q(z) \| p(z \mid x)\right) \end{aligned} \tag{3.9}$$

Since we know that the Kl divergence is always nonegative, then the ELBO is, as its name suggests, a lower bound of the evidence $\log p(x)$. That is,

$$\log p(x) \geq \text{ELBO}.$$

Furthermore, the gap between these quantities is precisely the discrepancy, in terms of the KL, between the approximate posterior $q(z)$ and the true posterior.

Remark 3.8.

When the latent variable is a parameter, the ELBO can be informally used for model selection, by finding the maximum a posteriori. However, when this is done in practice, it is unknown how far this solution is from the true one.

To do: include an illustration of the biased maxima

Remark 3.9.

Observe from eq. (3.7) that the first term in the ELBO is the objective of EM, this is because in EM we have that ELBO $= \log p(x)$, since $q(z) = p(z \mid x)$—see eq. (3.9). This is possible because EM is used in cases where $p(z \mid x)$ can be computed. In VI, however, we do not assume this, but rather we only consider a sufficiently good approximation q within the variational family $\mathcal{Q}$. Therefore, VI can be seen as an extension of EM, used in cases where the parameters are treated as random variables, but their posterior is intractable.

The mean field variational family. In practice, we need to choose an explicit family $\mathcal{Q}$. This is done considering the trade-off between the expressivity of the family and the feasibility of solving the problem in eq. (3.6). We will consider the mean field (MF) family, where the distribution over the latent variable $z = (z_1, z_2, \ldots, z_m) \in \mathbb{R}^m$ factorises as

$$q(z) = \prod_{i=1}^{m} q_i(z_i). \tag{3.10}$$

References

Andrieu, C., de Freitas, N., Doucet, A. & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*(1–2), 5–43.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction* (1st ed.). Cambridge, MA, USA: The MIT Press.