Lecture Notes

PROBABILISTIC GENERATIVE MODELS

This version: 9th 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, 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 statsitical prperties of the available observations. The *probabilistic* nature of the PGMs considered follows from the fact that the data generated 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 coruse, enforcing the sought-after PGM to have an explicit parametric pdf ca be rather restrictive.

Throughout the course, we will consider a probabilisty 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 will also consider the σ-algebra $\mathcal{F}$ to be the product Borel σ-algebra, that is, $\mathcal{F} = \mathcal{B}(\mathcal{X}) \otimes \mathcal{B}(\mathcal{Y}) \otimes \mathcal{B}(\mathcal{Z})$. This is the smallest σ-algebra making the joint random variable (X, Y, Z) measurable.

Furthermore, we will assume that the joint probability of (X, Y, Z) has a density, that is, $\forall A \in \mathcal{B}(\mathcal{X}), B \in \mathcal{B}(\mathcal{Y}), C \in \mathcal{B}(\mathcal{Z})$, we have

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

We will also assume that all marginals and conditional have a density. This includes $p(x,y)$, $p(y|x)$, $p(z|x,y)$, 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 need 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 much more intuitive distributions to model:

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

Therefore, the classifier is

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

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.

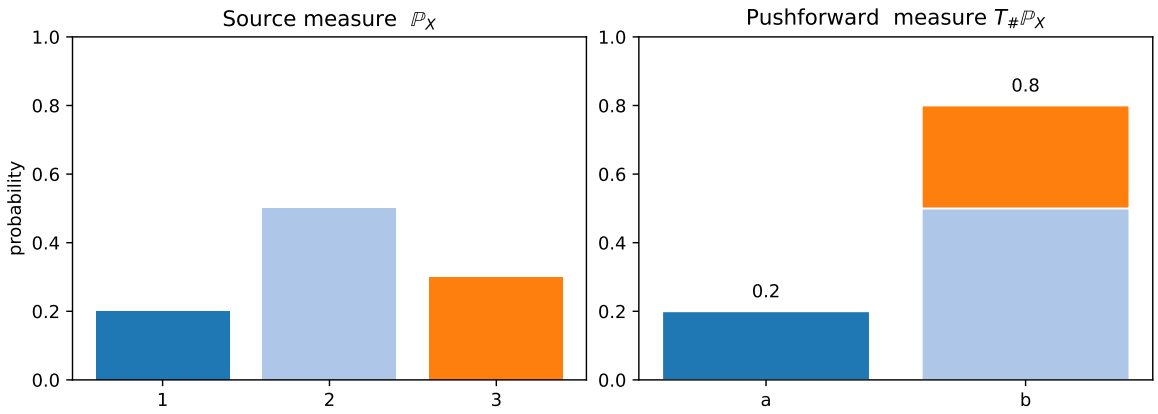

Figure 1.1: Source and pushforward distributions: discrete example.

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 see Fig. 1.2.

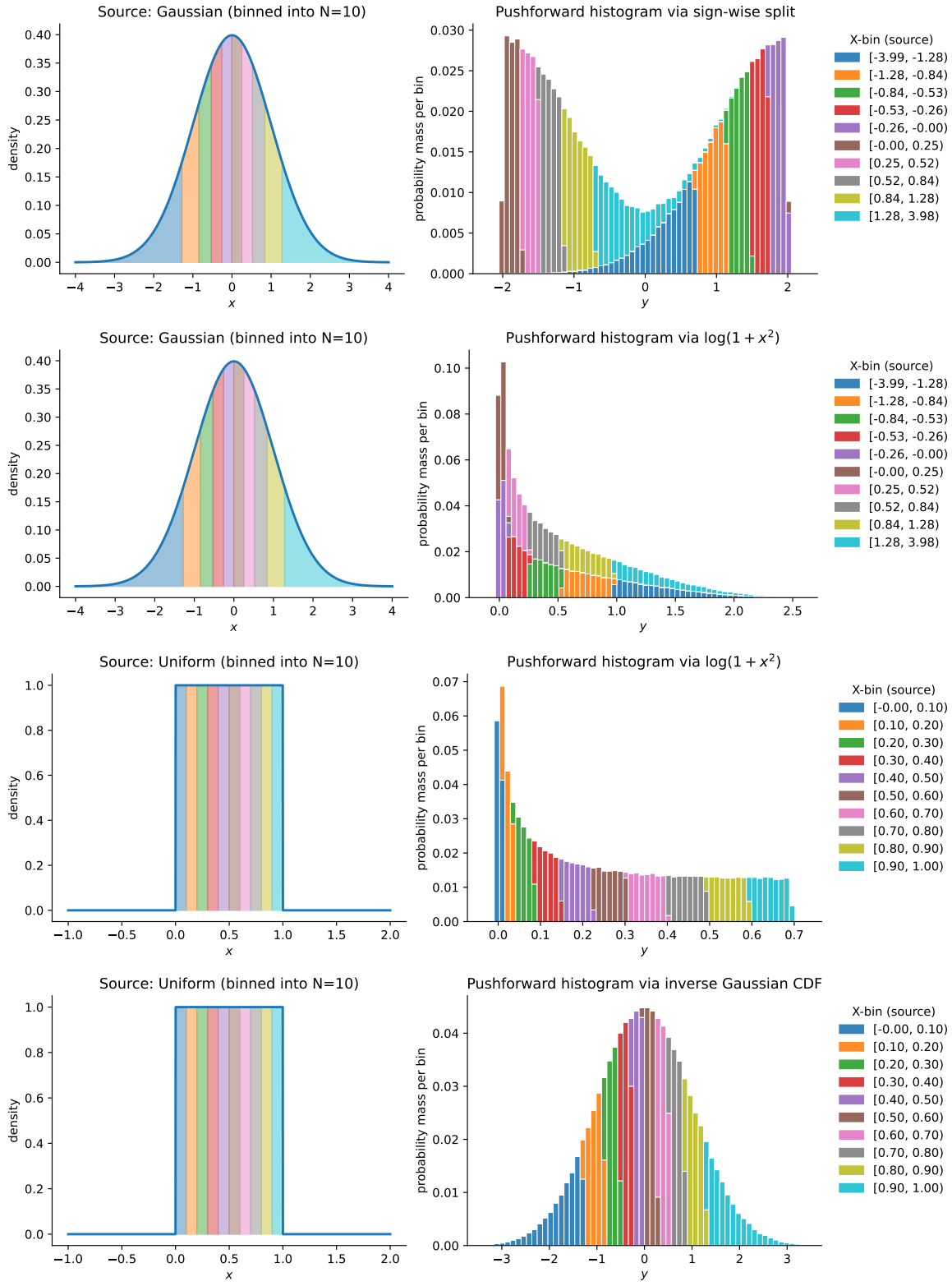

Figure 1.2: Source and target distributions: continuous examples

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

$$\mathbb{P}(T \in A) = \sum_{T(B_i)=A} \mathbb{P}(X \in B_i). \tag{1.6}$$

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,

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

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

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

Remark 1.4.
In general (but, importantly, not always) we will consider i.i.d observations, in which case the likelihood factorises as $L(\theta) = \prod_{i=1}^n p_\theta(y_i)$. Furthermore, when optimising the

likelihood we will consider the log-likelihood instead; in the i.i.d. case, this is

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{n} \log p(y_i). \tag{1.9}$$

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 $\{(x_i, y_i)\}_{i=1}^n$.

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

$$l(\theta) = \sum_{i=1}^{n} \frac{-1}{2} \log 2\pi\sigma^2 + \frac{1}{2\sigma^2}(y_i - ax_i)^2. \tag{1.11}$$

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

Remark 1.5.
Observe that optimising eq. (1.11) recovers the least squares solution.

Example 1.5 (Binary classification)**.**
Consider observations $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{0, 1\}$ 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 $s : \mathbb{R}^d \top \mathbb{R}$ is a feature extractor (e.g., $s(x) = a^\top x + b$). Assuming that the observations are i.i.d., we have

$$L(\theta) = \prod_{i=1}^{n} p(y_i|x_i) = \prod_{i=1}^{n} \sigma(s(x_i))^{y_i}(1 - \sigma(s(x_i)))^{1-y_i}, \tag{1.13}$$

and equivalently

$$l(\theta) = \sum_{i=1}^{n} y_i \log \sigma(s(x_i)) + (1 - y_i) \log(1 - \sigma(s(x_i))). \tag{1.14}$$

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

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

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

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 $\mu_k = x_i, \Sigma_k = 0$ which gives $l = \infty$.

We can overcome this drawback by introducing a collection of latent random variables Z_{nk}, 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?