

# Probability distributions and the Gaussian

Prof. Dr. Emre Onur Kahya · Department of Physics Engineering, Istanbul Technical University

Every method in this course either estimates a probability distribution or makes predictions that are best understood through one. This note collects the probability we need — how distributions are described, the four things we do with them, and the multivariate Gaussian in detail — and connects it to two pillars of modern machine learning: maximum-likelihood training of language models and Gaussian noise in diffusion models. Notation follows Note 1, Section 2.

## 1 Describing a distribution

A random variable  $X$  is described by how its probability is spread over its possible values. Three descriptions are in common use.

**Cumulative distribution function.** For a real-valued  $X$ , the c.d.f. is  $F(x) = P(X \leq x)$ . It is non-decreasing, tends to 0 as  $x \rightarrow -\infty$  and to 1 as  $x \rightarrow +\infty$ , and it exists for every random variable, discrete or continuous. For a vector  $\mathbf{X} = (X_1, \dots, X_d)$  the inequality is read component-wise:  $F(\mathbf{x}) = P(X_1 \leq x_1, \dots, X_d \leq x_d)$ .

**Probability table (discrete variables).** If  $X$  takes finitely or countably many values, its distribution is the list of probabilities  $P(X = x)$ , which are non-negative and sum to one. For several discrete variables the table becomes an array with one axis per variable; a joint table for  $d$  variables with  $m$  values each has  $m^d$  entries — a first sign of the difficulty Note 3 studies.

**Density (continuous variables).** If  $F$  is differentiable, the probability density function is  $p(x) = F'(x)$ , and probabilities of regions are integrals,

$$P(\mathbf{X} \in A) = \int_A p(\mathbf{x}) d\mathbf{x}, \quad \int_{\mathbb{R}^d} p(\mathbf{x}) d\mathbf{x} = 1.$$

A density is not a probability: it carries units of  $1/[x]$  and may exceed one. A Gaussian of width  $\sigma = 0.1$  mm has a peak density of about  $4 \text{ mm}^{-1}$ .

**The empirical distribution.** Given a sample  $x_1, \dots, x_n$ , the empirical c.d.f.  $\hat{F}(x) = \frac{1}{n} \#\{i : x_i \leq x\}$  is a step function that jumps by  $1/n$  at each data point. By the law of large numbers  $\hat{F}(x) \rightarrow F(x)$  for every  $x$ ; the convergence is even uniform in  $x$  (Glivenko–Cantelli). The empirical distribution is the simplest estimate of a distribution, and every learning algorithm can be viewed as a way of smoothing it.

## 1.1 Joint, marginal and conditional

For two variables with joint density  $p(\mathbf{x}, \mathbf{z})$  (sums replace integrals in the discrete case):

$$\begin{aligned} \text{marginal: } p(\mathbf{x}) &= \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}, \\ \text{conditional: } p(\mathbf{z} | \mathbf{x}) &= \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} \quad (p(\mathbf{x}) > 0), \\ \text{Bayes' rule: } p(\mathbf{z} | \mathbf{x}) &= \frac{p(\mathbf{x} | \mathbf{z}) p(\mathbf{z})}{\int p(\mathbf{x} | \mathbf{z}') p(\mathbf{z}') d\mathbf{z}'} \end{aligned}$$

Independence means  $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}) p(\mathbf{z})$ , equivalently  $p(\mathbf{z} | \mathbf{x}) = p(\mathbf{z})$ . Expectations are integrals against the density,  $\mathbb{E}[g(\mathbf{X})] = \int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$ ; the mean is  $\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}]$  and the covariance matrix is  $\Sigma = \mathbb{E}[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top]$ , with variances on its diagonal.

## 2 Four things we do with a distribution

Whether the distribution is a probability table, a Gaussian or a neural network with billions of parameters, the same four operations recur.

1. **Sample:** generate new data  $\mathbf{x} \sim p$ . This is what a generative model is for.
2. **Evaluate:** compute  $p(\mathbf{x})$  (the *likelihood* of a configuration), possibly after marginalising over unobserved variables. Low likelihood flags unusual data.
3. **Infer:** given some variables, predict others — the most probable value or the conditional mean  $\mathbb{E}[\mathbf{Z} | \mathbf{X} = \mathbf{x}]$ . Supervised learning is inference of  $Y$  from  $X$ .
4. **Learn:** choose the parameters  $\theta$  of a family  $p_\theta$  from data. The standard principle is *maximum likelihood*,

$$\hat{\theta} = \arg \max_{\theta} \prod_{i=1}^n p_\theta(\mathbf{x}_i) = \arg \min_{\theta} -\frac{1}{n} \sum_{i=1}^n \log p_\theta(\mathbf{x}_i), \quad (1)$$

the product being justified by the i.i.d. assumption. The right-hand side is the average negative log-likelihood, the density-estimation loss of Note 1, Table 2.

**Example 1** (a categorical variable). The number  $X \in \{0, 1, 2\}$  of reconstructed jets in a simplified event selection is described by three numbers  $\pi_0, \pi_1, \pi_2$  with  $\sum_k \pi_k = 1$ . *Sampling* draws  $u$  uniform on  $[0, 1)$  and returns the first  $k$  with  $\pi_0 + \dots + \pi_k > u$ . *Learning* by maximum likelihood from  $n$  events with counts  $n_0, n_1, n_2$  gives  $\hat{\pi}_k = n_k/n$ , the observed frequencies (Exercise E1).

### Modern connection — a language model is a categorical distribution

A large language model defines, for every prefix of text, a categorical distribution over a vocabulary of  $V \approx 10^5$  tokens. The network outputs scores (“logits”)  $z_1, \dots, z_V$  and the probabilities are  $\pi_k = e^{z_k/T} / \sum_j e^{z_j/T}$ , where the *temperature*  $T = 1$  during training. All four operations appear: training is maximum likelihood (1) on the next token, i.e. minimising cross-entropy; generating text is repeated sampling, with  $T < 1$  concentrating mass on likely tokens and  $T \rightarrow 0$  giving the single most probable token; evaluating the likelihood of a text gives its *perplexity*, exp of the average negative log-likelihood per token; and answering a question is inference, conditioning on the prompt. The example above — learned frequencies from counts — is exactly what such a model would do if it had no parameters to share between contexts.

### 3 The univariate Gaussian

The Gaussian (normal) density with mean  $\mu$  and variance  $\sigma^2$  is

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right). \quad (2)$$

It is a symmetric bump centred at  $\mu$  with width set by the standard deviation  $\sigma$  (Figure 1). About 68.3%, 95.4% and 99.7% of the probability lies within one, two and three standard deviations of the mean; physicists’ “5 $\sigma$ ” discovery threshold corresponds to a one-sided tail probability of  $2.9 \times 10^{-7}$ . Any Gaussian is a shifted and scaled standard one:  $X = \mu + \sigma Z$  with  $Z \sim \mathcal{N}(0, 1)$ .

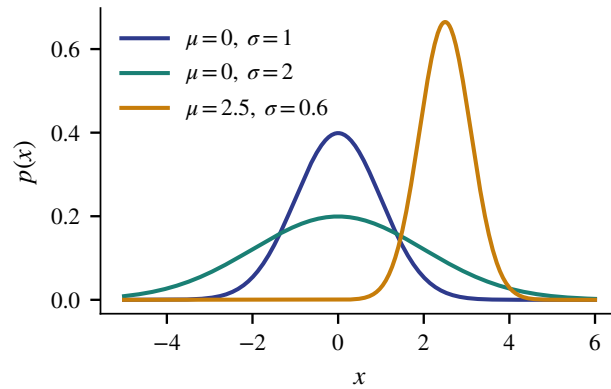


**Figure 1.** Gaussian densities. Doubling  $\sigma$  halves the peak height, since the area stays one.

Why this particular shape? The central limit theorem says that a sum of many independent contributions with finite variance is approximately Gaussian, whatever the individual distributions. Measurement noise built from many small effects is therefore often close to Gaussian. A second reason is that, among all densities with a given mean and variance, the Gaussian has maximal entropy: it assumes nothing beyond those two numbers.

**Gaussian noise and squared error.** Suppose targets are generated as  $y = f(x) + \varepsilon$  with  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ . Then  $-\log p(y | x) = \frac{(y - f(x))^2}{2\sigma^2} + \frac{1}{2} \log(2\pi\sigma^2)$ , and maximising the likelihood of a regression dataset over  $f$  is the same as minimising the sum of squared errors. The squared loss of Note 1, Table 2, is maximum likelihood under Gaussian noise; heavier-tailed noise calls for other losses (Exercise E3).

#### In physics — $\chi^2$ fits are Gaussian likelihoods

Fitting a model  $f_\theta$  to measurements  $y_i \pm \sigma_i$  by minimising  $\chi^2 = \sum_i (y_i - f_\theta(x_i))^2 / \sigma_i^2$  is maximum likelihood with independent Gaussian errors of known widths. The familiar rule “ $\Delta\chi^2 = 1$  gives the  $1\sigma$  interval” is the statement that  $-2 \log$  likelihood is quadratic near its minimum [2]. When errors are correlated, the sum becomes the quadratic form of the next section, with the covariance matrix of the measurements.

### 4 The multivariate Gaussian

For  $\mathbf{x} \in \mathbb{R}^d$ , the Gaussian with mean  $\boldsymbol{\mu} \in \mathbb{R}^d$  and symmetric positive-definite covariance matrix  $\Sigma \in \mathbb{R}^{d \times d}$  is

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \quad (3)$$

where  $|\Sigma|$  is the determinant. The prefactor only normalises the density; all the shape is in the exponent. The quantity

$$\Delta^2(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

is the squared *Mahalanobis distance* of  $\mathbf{x}$  from the mean: Euclidean distance after the data have been rescaled so that every direction has unit variance. Surfaces of constant density are the ellipsoids  $\Delta^2 = \text{const.}$

**Geometry through the eigendecomposition.** Write  $\Sigma = \sum_{i=1}^d \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$  with orthonormal eigenvectors  $\mathbf{v}_i$  and eigenvalues  $\lambda_i > 0$ . In the rotated coordinates  $u_i = \mathbf{v}_i^\top (\mathbf{x} - \boldsymbol{\mu})$  the exponent becomes  $\sum_i u_i^2 / \lambda_i$ , so the density factorises into  $d$  independent one-dimensional Gaussians with variances  $\lambda_i$ . The ellipsoid axes point along the eigenvectors with half-lengths proportional to  $\sqrt{\lambda_i}$ , and  $|\Sigma| = \prod_i \lambda_i$  measures the squared volume of the ellipsoid. Three families are common (Figure 2):

| Family    | Covariance                                   | Free parameters in $\Sigma$ | Shape                   |
|-----------|----------------------------------------------|-----------------------------|-------------------------|
| Isotropic | $\sigma^2 I$                                 | 1                           | spheres                 |
| Diagonal  | $\text{diag}(\sigma_1^2, \dots, \sigma_d^2)$ | $d$                         | axis-aligned ellipsoids |
| Full      | any positive-definite $\Sigma$               | $d(d+1)/2$                  | arbitrary ellipsoids    |

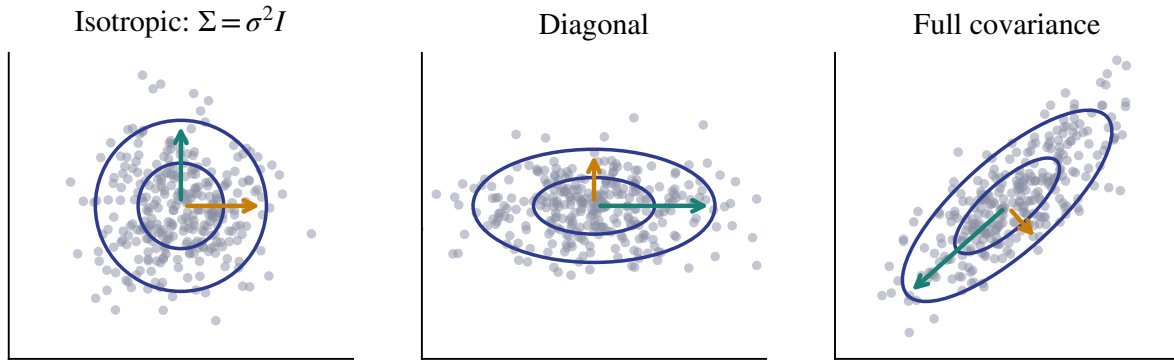


**Figure 2.** Samples (grey) and the one- and two- $\sigma$  contour ellipses of three Gaussians in  $d = 2$ . Arrows are the eigenvectors scaled by  $2\sqrt{\lambda_i}$ . The isotropic case has no preferred direction; the full covariance captures a correlation between the two coordinates.

The choice is a choice of model size. For  $d = 1000$  a full covariance has about half a million parameters, which  $n = 500$  examples cannot determine; Note 4 calls this a question of *capacity*.

**Closed under marginalising and conditioning.** Split  $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$  and partition  $\boldsymbol{\mu}$  and  $\Sigma$  accordingly. Every marginal of a Gaussian is Gaussian – just keep the corresponding entries,  $\mathbf{x}_a \sim \mathcal{N}(\boldsymbol{\mu}_a, \Sigma_{aa})$  – and every conditional is Gaussian,

$$\mathbf{x}_a \mid \mathbf{x}_b \sim \mathcal{N}(\boldsymbol{\mu}_a + \Sigma_{ab} \Sigma_{bb}^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b), \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba}). \quad (4)$$

The conditional mean is linear in the observed part and the conditional covariance does not depend on it. This is inference in closed form: if two detector channels are correlated, reading one sharpens the prediction for the other by an amount set by the correlation. Formula (4) underlies the Kalman filter used for particle-track reconstruction and spacecraft navigation, and Gaussian-process regression.

## 5 Sampling: from white noise to anything Gaussian

To sample  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ , factor  $\Sigma = LL^\top$  (Cholesky decomposition,  $L$  lower triangular), draw  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$  —  $d$  independent standard normals — and return

$$\mathbf{x} = \boldsymbol{\mu} + L\mathbf{z}. \quad (5)$$

Indeed  $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$  and  $\text{Cov}(\mathbf{x}) = L \mathbb{E}[\mathbf{z}\mathbf{z}^\top] L^\top = LL^\top = \Sigma$ , and a linear map of a Gaussian is Gaussian.

### Modern connection — Gaussian noise drives generative models

Equation (5) separates the randomness ( $\mathbf{z}$ ) from the parameters ( $\boldsymbol{\mu}, L$ ). Variational autoencoders exploit this “reparameterisation trick” to differentiate through sampling [4]. Diffusion models, behind many of today’s image generators, go further. Their *forward process* gradually turns data into noise: with  $\bar{\alpha}_t$  decreasing from 1 to 0,

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, I),$$

so  $\mathbf{x}_t | \mathbf{x}_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t)I)$ , an isotropic Gaussian (Figure 3). A neural network is trained to predict  $\mathbf{z}$  from  $\mathbf{x}_t$ ; running the process backwards from pure noise then generates new samples [3, 7]. Everything in this section — isotropic covariances, scaling and shifting, sampling by (5) — is used literally.

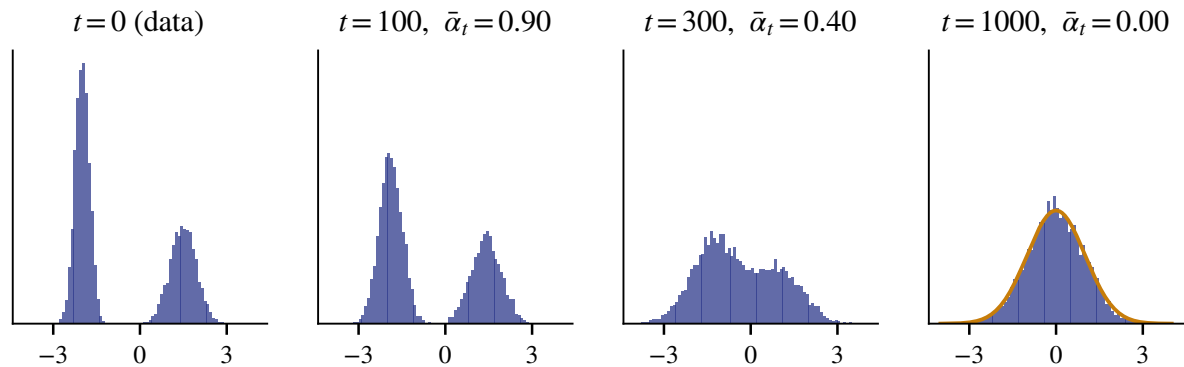


**Figure 3.** The forward process of a diffusion model applied to a one-dimensional, two-peaked data distribution (histograms of  $10^4$  samples, the noise schedule of Ho *et al.* [3] with  $T = 1000$  steps). By the last step the data are indistinguishable from  $\mathcal{N}(0, 1)$  (curve).

## 6 Learning the parameters

Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ , the maximum-likelihood estimates of a Gaussian are the empirical mean and covariance,

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top = \frac{1}{n} X_c^\top X_c, \quad (6)$$

where  $X_c$  is the design matrix of Note 1 with  $\hat{\boldsymbol{\mu}}$  subtracted from every row. In one dimension the derivation takes two lines: setting the derivatives of  $\sum_i [(x_i - \mu)^2 / (2\sigma^2) + \frac{1}{2} \log \sigma^2]$  with respect to  $\mu$  and  $\sigma^2$  to zero gives  $\hat{\mu} = \bar{x}$  and  $\hat{\sigma}^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2$ . The variance estimate is slightly biased,  $\mathbb{E}[\hat{\sigma}^2] = \frac{n-1}{n} \sigma^2$ , which is why statistics software often divides by  $n - 1$ ; for the sample sizes of machine learning the difference is negligible.

**When  $n$  is not much larger than  $d$ .** The matrix  $\hat{\Sigma} = X_c^\top X_c/n$  has rank at most  $n - 1$ , so for  $n \leq d$  it is singular: the fitted Gaussian is flat in some directions and (3) cannot be evaluated. Even for  $n$  somewhat larger than  $d$  its small eigenvalues are badly underestimated. The usual remedies are to restrict the family (diagonal or isotropic) or to *shrink*,  $\hat{\Sigma} + \epsilon I$  — the simplest instance of the regularisation discussed in Note 4.

**Computing log-densities.** In practice one evaluates  $\log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ , never the density itself, which underflows in high dimension. With  $\Sigma = LL^\top$ , solve the triangular system  $L\mathbf{u} = \mathbf{x} - \boldsymbol{\mu}$ ; then  $\Delta^2 = \|\mathbf{u}\|^2$  and  $\log|\Sigma| = 2 \sum_i \log L_{ii}$ . No matrix is ever inverted.

```
import numpy as np
from scipy.linalg import solve_triangular
def gaussian_logpdf(X, mu, Sigma):          # X: (n, d)
    L = np.linalg.cholesky(Sigma)
    U = solve_triangular(L, (X - mu).T, lower=True)    # (d, n)
    d = X.shape[1]
    log_det_half = np.log(np.diag(L)).sum()           # = 0.5 log|Sigma|
    return -0.5 * (U**2).sum(0) - log_det_half - 0.5 * d * np.log(2 * np.pi)
```

#### Modern connection — Mahalanobis distance as an anomaly detector

A trained classifier can be fooled by inputs unlike anything it has seen. A simple and strong defence fits a Gaussian to the network's internal feature vectors for each class (with a shared covariance) and flags a new input whose features have a large Mahalanobis distance to every class mean [5]. The same recipe — fit a Gaussian to normal operation, alarm on large  $\Delta^2$  — is used to monitor detectors and industrial processes; under the model,  $\Delta^2$  follows a  $\chi^2$  distribution with  $d$  degrees of freedom, which sets the alarm threshold.

**Summary** A distribution is given by a c.d.f., a probability table or a density; marginals, conditionals and Bayes' rule relate its parts. We sample from, evaluate, infer with and learn distributions, and maximum likelihood is the learning principle — for language models as for  $\chi^2$  fits. The multivariate Gaussian is described by a mean and a covariance whose eigenvectors and eigenvalues give the axes and widths of its elliptical contours. It is closed under marginalising and conditioning, sampled as  $\boldsymbol{\mu} + L\mathbf{z}$ , and fitted by the empirical mean and covariance, which require care when  $d$  is comparable to  $n$ .

## Exercises

- E1. Show that the maximum-likelihood estimate of a categorical distribution from counts  $n_1, \dots, n_m$  is  $\hat{\pi}_k = n_k/n$ . (Maximise  $\sum_k n_k \log \pi_k$  subject to  $\sum_k \pi_k = 1$  with a Lagrange multiplier.) What does it predict for a category never observed, and why is this a problem for a language model?
- E2. Verify that (2) integrates to one and has mean  $\mu$  and variance  $\sigma^2$ .
- E3. If the noise has the Laplace density  $p(\epsilon) = \frac{1}{2b} e^{-|\epsilon|/b}$ , which loss does maximum likelihood minimise? Which estimate of a constant  $\theta$  minimises  $\sum_i |y_i - \theta|$ , and why is it less sensitive to outliers than the mean?
- E4. For  $d = 2$ ,  $\boldsymbol{\mu} = 0$  and  $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$  with  $|\rho| < 1$ , find the eigenvalues and eigenvectors of  $\Sigma$  and

use (4) to compute  $p(x_1 \mid x_2)$ . How much does observing  $x_2$  reduce the variance of  $x_1$  when  $\rho = 0.9$ ?

- E5.** Let  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I_d)$ . Show that  $\mathbb{E} \|\mathbf{z}\|^2 = d$  and  $\text{Var} \|\mathbf{z}\|^2 = 2d$ . Conclude that for large  $d$  almost all samples lie near a sphere of radius  $\sqrt{d}$ , far from the mode at the origin. (Note 3 returns to this.)
- E6.** Implement sampling by (5) and `gaussian_logpdf`. Draw  $n$  samples from a Gaussian in  $d = 50$  with a random full covariance, refit by (6), and plot the average test log-likelihood against  $n$  for the full, diagonal and shrunk ( $\hat{\Sigma} + \epsilon I$ ) estimates. Which wins for small  $n$ ?

**Further reading.** Bishop [1], sections 2.1–2.3; Murphy [6], chapters 2–3; for data analysis in physics, Cowan [2].

## References

- [1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer (2006).
- [2] G. Cowan, *Statistical Data Analysis*, Oxford University Press (1998).
- [3] J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, *NeurIPS* (2020).
- [4] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, *ICLR* (2014).
- [5] K. Lee, K. Lee, H. Lee and J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, *NeurIPS* (2018).
- [6] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*, MIT Press (2022).
- [7] Y. Song *et al.*, Score-based generative modeling through stochastic differential equations, *ICLR* (2021).

The selection and order of topics in weeks 1–3 follow the course IFT 6390 *Fondements de l'apprentissage machine* at Université de Montréal (taught by Ioannis Mitliagkas, slides by Pascal Vincent). The text, examples, figures and exercises of this note are original. Figures are computed from synthetic data by `figures.py`, published with the sources of these notes.