E178/ME292b
Statistics and Data Science for Engineers
Reader

Gabriel Gomes

September 25, 2023

Contents

Symbols

Sets

$\{\}$	... the empty set
$\mathbb{R}$	... the real numbers
$\mathbb{R}^+$	... the positive real numbers, not including zero
$\mathbb{R}^D$	... D-dimensional vector space of real numbers
$\mathbb{Z}$	... the integers
$\mathbb{N}$	... the natural numbers not including zero
$\mathbb{N}_0$	... the natural numbers including zero
$\{a_i\}_n$	... the set $\{a_1, a_2, \ldots, a_n\}$, indexed with i.
$a \in A$	... a is a member of the set A.
$\forall a \in A$	... indicates that a condition holds for all elements a in the set A
$A \cup B$	... the union of sets A and B
$A \cap B$	... the intersection of sets A and B
$\cup_{i=1}^{n} e_i$	... the union of sets $e_1, e_2, \ldots, e_n$.
$A \subset B$	... A is a strict subset of B; i.e. all elements of A are also elements of B, and $A \neq B$.
$A \subseteq B$	... A is a subset of B, and A is possibly equal to B.
$A \setminus B$	... The set that results from removing from A all of the elements of B.
(a, b)	... open interval. All real numbers between a and b, excluding a and b.
$[a, b]$	... closed interval. All real numbers between a and b, including a and b.
$(a, b]$	... All real numbers between a and b, excluding a and including b.
$[a, b)$	... All real numbers between a and b, including a and excluding b.

Functions

$f : A \to B$	... f is a function from set A to set B. The inputs of f are elements of A and its outputs are elements of B.
$f(x; \theta)$	... f is a function with inputs x and parameters θ

Probability

Y	... A random variable
Ω_Y	... Sample space of Y
$\mathcal{E}_Y$	... Event space of Y
P_Y	... Probability measure of Y
p_Y	... Probability density function of Y
Φ_Y	... Cumulative distribution function of Y
$E[Y]$, μ_Y	... Expected value (mean) of Y
$Var[Y]$, σ_Y^2	... Variance of Y
σ_Y	... Standard deviation of Y
$Cov(X, Y)$	... Covariance of X and Y
$y \sim Y$	... y is a sample of Y
$Y \sim p_Y$	... Y has pdf p_Y
$\mathcal{B}(\alpha)$	... Bernoulli distribution with parameter α
$\mathcal{U}(a, b)$	... Uniform distribution on the interval $[a, b]$
$\mathcal{N}(\mu, \sigma^2)$	... Normal distribution with mean μ and variance σ^2
$\{y_i\}_N \overset{\text{iid}}{\sim} Y$	... $\{y_i\}_N$ is iid sampled from Y
$\{Y_i\}_N \overset{\text{iid}}{\sim} Y$	... $\{Y_i\}_N$ are iid copies of Y
$Y \vert X = x$	... The random variable Y conditioned on $X = x$
ρ_{XY}	... Correlation between random variables X and Y

Statistical learning

$\mathcal{H}$	... A family of prediction functions
P	... The number of parameters that characterize $\mathcal{H}$
$\underline{\theta}$	... The vector of parameters $\theta \in \mathbb{R}^P$
$h(x; \underline{\theta})$	... A family of prediction functions

Overview of the course

Much of the activity of engineers involves using *models* to predict and control the behavior of physical systems. We use models of solar panels and batteries to design solar farms and to predict their production. We use models of the drivetrain of a car to design cruise control systems. We model the water pressure in a pipe in order to avoid cavitation. Models are also used in real-time feedback control loops to guide systems that operate in uncertain environments. An example of this is "model predictive control", which is a technique used in robotics and other fields.

To build a model of a system is to specify a function that maps the "inputs" of the system to its "outputs". But what do we mean here by "inputs" and "outputs"? These are quantities that, at the very least, must relate somehow to measurements that we can gather. Take for example a quadcopter. The measurable quantities include the voltages applied to each of its motors, its position, its linear and angular speeds, and the masses and moments of inertia of its parts. We regard some of these as "inputs" (the voltages), others as "outputs" (positions and speeds), and others as "parameters" (masses, and moments of inertia). Perhaps we assign these labels according to our intuitive sense of causality. But more importantly, we designate the positions and velocities as outputs because they are the quantities that we wish to steer. The voltages are the inputs because we can manipulate them more or less at will, and we notice that they influence the outputs in a predictable way. The masses and moments of inertia are parameters of the model because they mediate the relationship between the inputs and the outputs.

Having decided which measurements are inputs and outputs, we can go on to build a model. There is a large number of modeling options and techniques to choose from. Figure 1 shows one way of conceiving the space of possible models: arranged on a spectrum according to their reliance on data versus a-priori principles. On the extreme left-hand side we have models with no a-priori structure, meaning that they make no use of mechanistic principles such as Newton's laws, conservation laws, or the physical laws of electricity, thermodynamics, fluid mechanics, etc. Rather, they acquire their structure directly from the data. These data-centric models are the focus of this course. Moving from left to right we encounter models with increasing a-priori structure, and all the way to the right we find the "mechanistic models", which are ones that get their structure primarily from a-priori principles. These are the focus of most other modeling courses in the engineering curriculum.

There is a trade-off between the amount of a-priori structure in a model and its number of tunable parameters. A model with more structure will typically have fewer parameters,

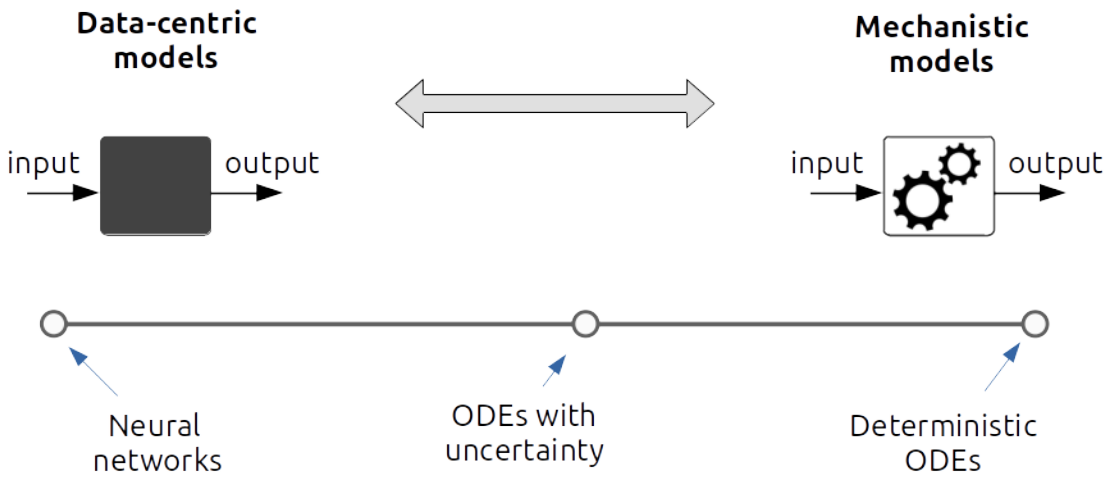

Figure 1: Models range from purely data-based to purely mechanistic

and will therefore require less data to calibrate (a.k.a. to train). The highly structured models on the right of the diagram have only a few parameters to tune. The formula below for the motion of a quadcopter is an example.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} n_1/J_{xx} \\ n_2/J_{xx} \\ n_3/J_{zz} \end{bmatrix} - \frac{J_{xx} - J_{zz}}{J_{xx}} \begin{bmatrix} -qr \\ pr \\ 0 \end{bmatrix} \tag{1}$$

In this formula, p, q, and r are rotational velocities, n_1, n_2, and n_3 are torques, and J_{xx} and J_{zz} are two scalar moments of inertia. The formula was derived using the principles of rigid body dynamics and a description of the geometry of the quadcopter[1]. In this sense it is an open-box model – because to build it we were allowed to peek inside and observe the inner workings of the system. The closed-box models at the opposite end of the spectrum rely only on the measurements of the inputs and outputs for their construction.

Both types of models can serve the purpose of predicting how the system will behave under given inputs. In the case of Eq. 1, this prediction is calculated by integrating the differential equations forward in time, using a given initial condition and the input sequence. This is a "deterministic" model, in the sense that the calculation yields the same answer every time it is executed. But this does not mean that the model is devoid of error. Even though the kinematic and dynamic principles that were used to derive the formula are extremely solid, there will always remain some *uncertainties*: aspects of the system whose details remain obscure and unmodeled. For example, if our quadcopter were removed from its cozy lab environment and operated outdoors, it would be subjected to unpredictable winds and other disturbances. Similarly, properties such as the friction coefficients may change over time in ways that we cannot anticipate. The measurements themselves have only finite

[1]The derivation itself is not important here, but if you are interested I recommend the course "ME136/236U" Control and Dynamics of Unmanned Aerial Vehicles".

precision and introduce their own bit of uncertainty. Thus, no model is perfect. Whether and how to characterize the uncertainty of a system is a modeling choice.

Between the two extremes we find techniques that combine a-priori principles for the open parts with uncertainty modeling for the closed parts. For example, in the case of the quadcopter, we might speculate that the measurements of angular speed have errors that are equally likely to be positive or negative. This would be modeled with a *measurement equation*:

$$\begin{bmatrix} p_m \\ q_m \\ r_m \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \epsilon_p \\ \epsilon_q \\ \epsilon_r \end{bmatrix} \tag{2}$$

Here, (p, q, r) are the true rotational velocities, (p_m, q_m, r_m) are the measurements, and $(\epsilon_p, \epsilon_q, \epsilon_r)$ are *random variables* whose numerical values may change each time the formula is evaluated. The structure of a random variable is encoded in its *distribution*, which is a function that assign probabilities to each of its possible values. In our quadcopter example, we might assign to each of the ϵ's a Gaussian (a.k.a. *normal*) distribution with zero mean and a given standard deviation. Our assumption that the errors are evenly distributed around zero is captured by setting the mean of the Gaussians to zero. The advantage of doing this is that we can now leverage the additional information (the standard deviations) to compute a more precise estimate of the true rotational velocities using *estimation* (a.k.a. *filtering*), for example using an extended Kalman filter or a particle filter. These techniques, along with others such as Monte Carlo simulation and robust control, fall toward the middle of the spectrum of Figure 1.

Equation 1 is but one of many possible models of quadcopter motion. It is a *dynamical* model because it maps input functions of time to output functions of time. To do this, it requires a specification of the initial state of the system. *Static* models on the other hand are instantaneous maps. Each input value maps uniquely to an output values, which does not depend on the state of the system. Which type of model to use – static or dynamic – is a modeling choice that depends largely on the time scale of interest, and how it compares to the natural time scale of system. In the quadcopter example, the time scale of Eq. 1 is on the order of seconds, since it takes only a few seconds for the quadcopter to reach maximum speed under full thrust. This level of detail is needed if our objective is to navigate a complex terrain at high speed. However, if our goal is simply to estimate the flight time for a long journey, on the order of tens of minutes, then it makes more sense to ignore the dynamics and use a static model. That is, to compute flight time by dividing the travel distance by the average speed.

The majority of this course will focus on *static closed-box* models, although we will also briefly introduce *dynamical closed-box* models in Chapter **??**. We will begin in Chapters 1 and 2 by establishing some mathematical preliminaries in probability theory and optimization theory. Then in Chapter 3 we introduce the simplest case: models of static systems with no inputs (left side Figure 2). Measurement errors are an important example of this case, since they can be modeled as random numbers generated by a certain distribution, and with no influence from any other measured quantity. The main problem of interest here is the

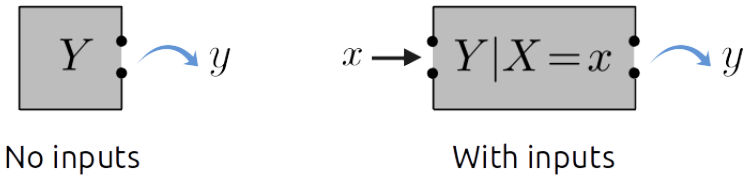

Figure 2: Two cases of closed-box models. The capital letters in the box are random variables. The one on the right is a "conditional" random variable (Y given X equals x). We will learn about these in Chapter 1.

inference problem, which is to deduce properties of the hidden process from the observed data. This is an important problem in its own right, with many real-world applications. But it also serves to introduce concepts that will reappear in the input/output case. Concepts such as point estimates, confidence intervals, hypothesis tests, maximum likelihood, mean squared error, etc.

Chapter 4 introduces the case of static models *with* inputs. Many techniques in machine learning fall into this category, and indeed the topic will occupy most of the second half of the course, with the exception of Chapter **??** on *dynamical* models with inputs. We will begin by describing a second type of problem that arises in the input/output context: the *prediction* problem. This problem asks how to create a function (i.e. a model) that predicts the output of a system given its input? The general paradigm for solving this is called *supervised learning* and is the topic of Chapter 4. In this chapter we will encounter several important concepts that are common to all of the various approaches to input/output modeling, including regression vs. classification, cross-validation, hyper-parameters, overfitting, and others.

With this general understanding in place, we will be in a good position to tackle several specific types of models that are used in machine learning applications. These include linear regression and briefly K-bins and K-nearest neighbors in Chapter **??**, naive Bayes and logistic regression in Chapters **??** and **??**, neural networks in Chapter **??**, decision trees and ensemble methods in Chapter **??**, and (time permitting) support vector machines in Chapter **??**. Apart from these, we will also briefly study the statistical approach to dynamical (a.k.a. time series) models in Chapter **??**.

Chapter 1

Probability theory

Probability theory is the study of *uncertainty*. Most things in life are uncertain. Predictions about the future are uncertain because they are subject to many influences that we do not fully know or control. Statements about the present are also uncertain, due to the finite precision of our measurement devices. We use probability theory to quantify, combine, and evolve interacting uncertainties. This helps us to get a sense of the confidence that we can place on statements about the world. Large uncertainty means low confidence; low uncertainty means high confidence.

A *probability* is a number between 0 and 1 that we assign to an *event*. An event is, loosely speaking, anything that happens in the world. We will give this term a more precise definition soon, but first let's consider the semantic question: what do we mean when we say that something has "a probability of 0.5"?

Here is a statement: "The temperature outside my office is between 60°F and 65°F". This statement is either true or false, but I do not know which because I have not yet taken a measurement with a thermometer. However my *belief* is that it is false, based on what I see through my window. If asked to rate my belief on a scale from 0 to 1 – where 0 means complete certainty that the statement is false and 1 means complete certainty that it is true – I would give it a 0.2. It is less than 0.5 because I lean toward 'false', but I am not certain. This is the so-called *Bayesian* interpretation of probability. The probability value quantifies our subjective belief in a proposition. Whenever a belief is based on measurements or perceptual experience, then its probability (or *credence* in the Bayesian terminology) must lie in the open interval (0,1). That is, 0 and 1 are disallowed. Complete certainty is reserved for mathematical statements such as "there is no largest prime number", which are made in the context of a set of axioms that are *defined* to be true.

Aside from quantifying belief, we can also use probabilities to gauge our uncertainty about the outcome of a process or measurement. Take for example the tossing of a coin. When we say that the "probability of heads is 0.5", we mean that, if the coin were tossed a large number of times, we expect it would turn up heads about half of the time. More precisely we mean that as the number of trials is increased, the ratio of heads to the total number of tosses will eventually approach 0.5. We cannot say with any certainty what the sequence of heads and tails will be, but we are positive that the ratio will converge to 0.5 –

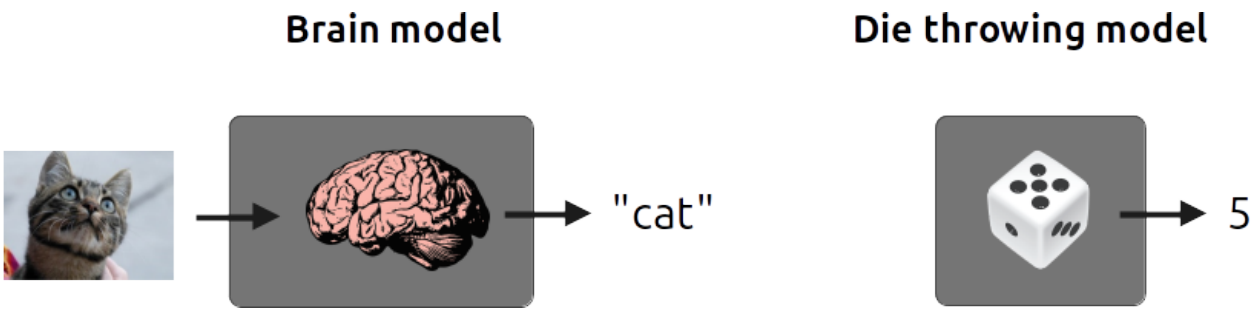

Figure 1.1: Two closed-box models

provided the coin is fair. This is the *frequentist* interpretation of probability.

Neither of these interpretations, the Bayesian nor the frequentist, are completely satisfactory because they do not suggest a practical method for their measurement. Psychologists have made progress in the design of experiments that quantify subjective beliefs, however it remains a difficult problem. On the other hand, the frequentist definition relies on an infinite experiment, and that is also difficult! Despite this, the theory of probability has been extremely successful in modeling real-world uncertainties of both types. Furthermore, the mathematics of probability theory applies equally to both interpretations. Thus, we can proceed without worrying too much about the interpretation, but keeping in mind that both are available.

1.1 Sample space, events, and event space

We've defined a closed-box system as one that accepts and produces data, but whose inner workings are unknown. The brain is often modeled as a closed box. It is conceivable that some day we will be able to build a mechanistic model of a human brain, or a significant portion thereof, and furthermore that we will have the computing power to run the model (i.e. to simulate the brain) in some reasonable amount of time. Today however we can only build closed-box models, such as neural networks, that learn from data to behave in a brain-like fashion, in some very restricted sense. Figure 1.1 shows a simple diagram of such a model. It receives a photo as input, and produces a determination of whether the photo shows a cat.

The roll of a die is another example. Again, it may be possible to produce a mechanistic model of the die based on 3D rigid-body dynamics, interacting with the fluid dynamics of the air. However this model would require inputs such as the initial translational and rotational velocities of the die, the fluid properties of the air, the coefficients of friction and restitution of the floor, etc., which may be so uncertain as to render the predictions worthless. The closed-box model for the die is very simple: with each evaluation it produces a number, 1 through 6 with equal probability. This particular model has no inputs.

The *sample space* of a model is the set of all possible values of its output, or its *outcomes*. In the case of temperature, the sample space is the real line $\mathbb{R}$. This might seem strange,

since not all real numbers are possible temperatures. Why include -1 Kelvin in the sample space? We dispense with this problem by assigning zero probability to impossible outcomes.

The sample space for the rolled die has six elements: {1, 2, 3, 4, 5, 6}. This is a *discrete* sample space, as opposed to the real line which is *continuous* (in the sense of 'continuous variable'). Both of these sample spaces are *numerical*. The sample space of the brain system of Figure 1.1 is *categorical* because it consists of *labels* 'cat' and 'no cat'. The difference between these two types is that numerical sample spaces have an inherent order, whereas categorical sample spaces do not.

A *random variable* is a symbol that represents some uncertain quantity. In our examples, the outside temperature, the brain utterance, and the outcome of the die are random variables. Random variables are typically (but not always) denoted by upper-case Roman letters. T for the outside temperature, R for the roll of a die, and B for cat identification, for example. We use Ω for the sample space, and indicate its random variable with a subscript.

$$\Omega_T = \mathbb{R} \tag{1.1}$$
$$\Omega_R = \{1, 2, 3, 4, 5, 6\} \tag{1.2}$$
$$\Omega_B = \{\text{'cat'}, \text{'no cat'}\} \tag{1.3}$$

An *event* e is any subset of the sample space: $e \subseteq \Omega$. The statement "the temperature outside my office is between 60°F and 65°F" corresponds to the event $e = [60, 65]$. We can also denote this event as $T \in [60, 65]$. "The temperature outside my office exceeds 80°F" corresponds to the event $e = (80, \infty)$, and can be written as $T > 80$. Similarly for events in discrete sample spaces:

- $e_1 = \{1, 2\}$ is the event "roll less than a 3".
- $e_2 = \{1, 2, 3, 4, 5, 6\}$ is the event "roll any number".

Each time we roll the die, it produces an outcome. If we roll a 2, then both events e_1 and e_2 are observed. If we roll a 3, then only e_2 is observed.

The *event space* is the set of all possible events, i.e. all subsets of Ω [1]. This is a much larger set than Ω, known as its *power set*. If we use $|\Omega|$ for the size of (i.e. the number of elements in) the sample space, and $|\mathcal{E}|$ for the size of the event space, then

$$|\mathcal{E}| = 2^{|\Omega|} \tag{1.4}$$

1.2 Probability measure

A *probability measure* P is a function that assigns a real number to each event in an event space.

$$P : \mathcal{E} \to \mathbb{R} \tag{1.5}$$

[1] Actually we do not have to include *all* of the subsets in the event space, only enough to form a "σ-algebra". That is, Ω must contain the complements of each of its elements, as well as the intersections and unions of any number of its elements.

To qualify as a probability measure, the function must satisfy the following three properties, known as the *axioms of probability*.

A1. All probabilities are non-negative.

$$P(e) \geq 0 \qquad \forall e \in \mathcal{E} \tag{1.6}$$

A2. The probability of the sample space is 1.

$$P(\Omega) = 1 \tag{1.7}$$

A3. For any *disjoint* set of events $\{e_i\}_n$, meaning that no two events occur simultaneously ($e_i \cap e_j = \{\}$ whenever $i \neq j$), the probability that any occur equals the sum of the probabilities that each occur.

$$P(\cup_{i=1}^{n} e_i) = \sum_{i=1}^{n} P(e_i) \tag{1.8}$$

These axioms were stated in the 1930's, much after the initial development of probability theory in the sixteenth century. They capture our intuitions for both the Bayesian and frequentist notions of probability, and they are a sufficient foundation for the full development of the theory.

The following properties are easily deduced from the axioms. I encourage you to try to prove them.

1. Nothing can't happen:
$$P(\{\}) = 0 \tag{1.9}$$

2. If e' happens whenever e happens, then the probability of e cannot exceed that of e':
$$e \subseteq e' \;\Rightarrow\; P(e) \leq P(e') \qquad \forall e, e' \in \mathcal{E} \tag{1.10}$$

3. Everything either happens or it doesn't:
$$P(e) + P(\Omega \setminus e) = 1 \qquad \forall e \in \mathcal{E} \tag{1.11}$$

4. The probability of either e or e' happening equals the sum of their probabilities, minus the probability that they both happen:
$$P(e \cup e') = P(e) + P(e') - P(e \cap e') \qquad \forall e, e' \in \mathcal{E} \tag{1.12}$$

Example 1.2.1. A scale is used on a production line to monitor the weight of the widgets produced. It finds that 30% weigh less than 120 g, 40% weigh more than 200 g, and 50%

weight between 120 g and 250 g. Describe this situation using a random variable and a probability measure. What percentage of widgets weigh between 200 g and 250 g?

Solution. We define a random variable W with sample space $\Omega_W = \mathbb{R}$. Figure 1.2 shows

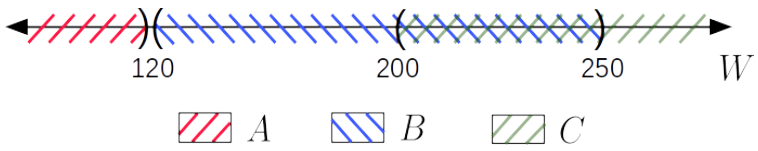

Figure 1.2: Example 1.2.1

the three defined events:

$$A = (-\infty, 120) \quad \text{with } P(A) = 0.3 \tag{1.13}$$
$$B = (120, 250) \quad \text{with } P(B) = 0.5 \tag{1.14}$$
$$C = (200, \infty) \quad \text{with } P(C) = 0.4 \tag{1.15}$$

Our goal is to find the probability of event $E = (200, 250)$. Using Eq. 1.12 we find,

$$P(A \cup B \cup C) = P(A \cup B) + P(C) - P((A \cup B) \cap C) \tag{1.16}$$

Since $A \cup B \cup C = \Omega$, and also $E = (A \cup B) \cap C$, this means that,

$$1 = P(A \cup B) + P(C) - P(E) \tag{1.17}$$

Using the third axiom we find that $P(A \cup B) = P(A) + P(B) = 0.8$, and therefore $P(E) = 0.8 + 0.4 - 1 = 0.2$.

1.3 The probability density function

The probability measure P returns the probabilities for all of the possible events of an experiment. However it is not a convenient object for practical use. To program a probability measure, one would have to write a function that accepts every possible subset of the sample space and returns a non-negative number for each one. Without some simplifying property or rule, this would require storing the set of all possible events, which as we have seen, grows exponentially with the size of the sample space.

Fortunately the axioms of probability ensure the existence of another function that captures the same information but is simpler to use. This is the *probability density function* (pdf), or the *distribution* of the random variable. The pdf is simpler because it maps the *sample space* (as opposed to the event space) to the reals. We denote the pdf with lower

case p, with a subscript indicating its random variable:

$$p_T : \Omega_T \to \mathbb{R} \tag{1.18}$$
$$p_R : \Omega_R \to \mathbb{R} \tag{1.19}$$

The defining property of the probability density function is that its integral over any event e equals the probability of e.

$$P(e) = \int_e p(\omega) d\omega \qquad \forall e \in \mathcal{E} \tag{1.20}$$

Or, if the sample space is discrete, it is the sum over the elements of e:

$$P(e) = \sum_{\omega \in e} p(\omega) \tag{1.21}$$

Properties of probability density functions
The two properties below can be regarded as axioms for probability density functions.

1. Non-negativity:
$$p(\omega) \geq 0 \qquad \forall\, \omega \in \Omega \tag{1.22}$$

2. Sum to one:
$$\int_\Omega p(\omega) d\omega = 1 \qquad \text{(continuous sample space)} \tag{1.23}$$
$$\sum_{\omega \in \Omega} p(\omega) = 1 \qquad \text{(discrete sample space)} \tag{1.24}$$

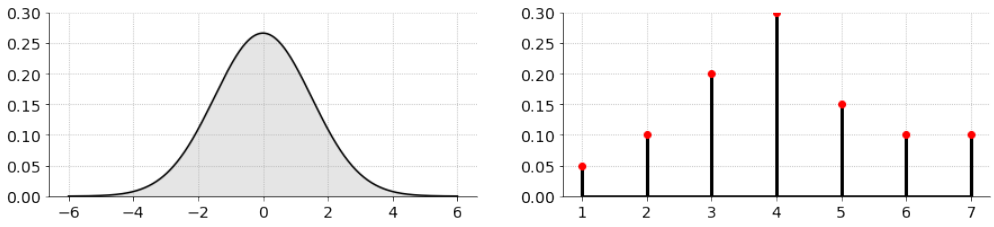

Figure 1.3: Example probability density functions.

A note on integrals and sums Figure 1.3 shows probability density functions over continuous and discrete sample spaces. Many introductory texts on probability theory refer to the discrete version as a probability *mass* function. However, in this course we will dispense with this distinction and refer to both as probability density functions. We will also use integrals only, no sums. This should not create confusion. If the sample space is discrete, then the integral should be interpreted as a sum. This does not violate any rules of mathematics, since we take the integrals to be of the Lebesgue type.

Example 1.3.1. Let X be a continuous random variable with $\Omega_X = [1, \infty)$. The pdf of X is of the form $p_X(x) = a\, x^b$, where a and b are integers. Find the conditions that must hold on a and b.

Solution. Non-negativity requires that $a\, x^b \geq 0$ for all $x \in [1, \infty]$. Thus we conclude $a \geq 0$. Secondly, we require that the integral of $p_X(x)$ over the sample space equal 1. The integral is only defined when $b < -1$, so we adopt that assumption.

$$\int_1^\infty a\, x^b = \frac{a}{b+1} x^{b+1} \Big|_1^\infty = \frac{a}{b+1}(0-1) = -\frac{a}{b+1} = 1 \tag{1.25}$$

From which we obtain a third condition: $a + b = -1$.

Two questions that often arise about data are a) what is its central tendency or average, and b) how widely do the values spread. These are captured respectively by the *expected value* (or *mean*) of the random variable, and by its *variance*.

1.4 Expected value

The expected value of a random variable Y, also known as the *expectation* or the *mean* of Y, is denoted with $E[Y]$ or μ_Y and is defined as follows,

$$E[Y] = \int_{\Omega_Y} y\, p_Y(y)\, dy \tag{1.26}$$

As in Eq. **??**, the integral should be interpreted as a sum if Y is discrete-valued. The expected value can be understood as the "balance point" of the pdf. If we cut a piece of paper into the shape of the pdf, then this shape will be balanced by a fulcrum at $E[Y]$, as shown in Figure 1.4. The figure also illustrates the "median", which is any point y that satisfies $P(Y \leq y) = 0.5$ and $P(Y \geq y) = 0.5$. That is, any point for which there is equal probability of sampling above or below it. In a symmetric distribution such as distribution A on the left hand side of Figure 1.4, the median coincides with the mean, and they are both at the point of symmetry. We can appreciate a difference between the median and the mean if we convert distribution A into distribution B (on the right hand side) by taking a portion of the high-value outcomes and moving them further to the right. Distribution B is said to be "positively skewed", or "right skewed", since the action causes the mean (the balance point) to also move to the right. However it does not affect the median, since the areas to its left and right remain unchanged. Figure 1.5 further generalizes this notion. It shows positively and negatively skewed distributions and a symmetric distribution, along with their respective medians and means. The mean of the positively skewed distribution is to the right of its median, and conversely for the negatively skewed distribution. This is often but not always the case. The skew of a distribution has a mathematical definition that beyond our scope, and it is not always true that a distribution with positive skew has a mean that is greater than its median. We will not pursue the median or the skew of a distribution further in this course.

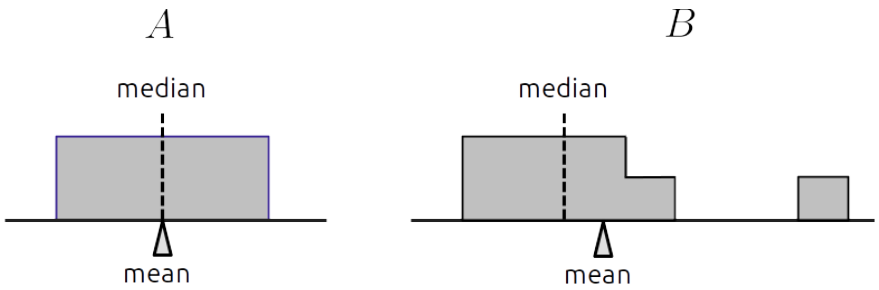

Figure 1.4: Means vs. median. B is obtained by moving a portion of distribution A to the right. This action affects the mean but not the median.

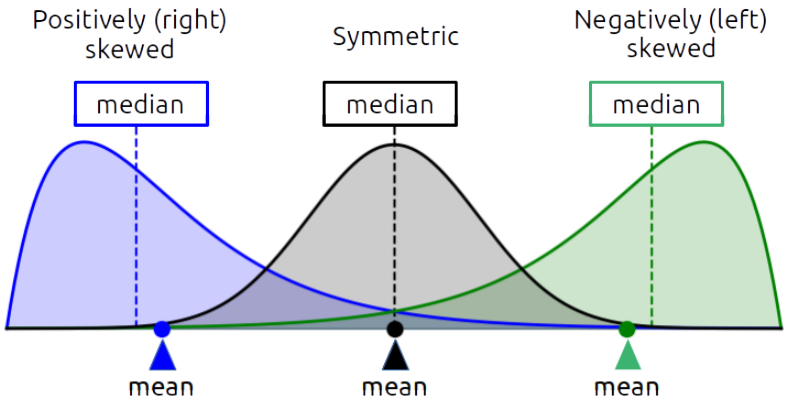

Figure 1.5: Right, center, and left skew

Properties of the expected value

The properties below follow from the definition of the expected value.

1. $E[\cdot]$ is a *linear* operation. This means that the expectation of a linear combination of random variables $\{Y_i\}_n$ (all over the same sample space) equals the linear combination of their expectation.
$$E\left[\sum_{i=1}^{n} \alpha_i Y_i\right] = \sum_{i=1}^{n} \alpha_i E[Y_i] \tag{1.27}$$

2. The expected value of a fixed number equals that number: $E[\alpha] = \alpha$.

3. The expected value of a function g of a random variable Y is computed with,
$$E[g(Y)] = \int_{\Omega_Y} g(y)\, p_Y(y) dy \tag{1.28}$$

Example 1.4.1. Find the expected value of the distribution of Example 1.3.1.

Solution. In the example we found that $a+b = -1$, so the pdf is of the form $p_X(x) = ax^{-1-a}$. Next we apply the definition of the expected value.

$$E[X] = \int_1^\infty x\, a\, x^{-1-a} dx \tag{1.29}$$
$$= a \int_1^\infty x^{-a} dx \tag{1.30}$$

The integral exists only if $a > 1$. Then

$$E[X] = \frac{a}{1-a}\, x^{1-a}\Big|_1^\infty \tag{1.31}$$
$$= \frac{a}{a-1} \tag{1.32}$$

1.5 Variance

The *variance* of a random variable Y is denoted with $Var[Y]$ or σ_Y^2, and is defined as

$$Var[Y] = E[(Y - E[Y])^2] \tag{1.33}$$

Here, $(Y - E[Y])^2$ is a random variable, a sample of which is obtained by squaring the distance from a sample of Y to the fixed mean $E[Y]$. The expected value of this squared distance is the variance of Y. An alternate formula for the variance can be derived by

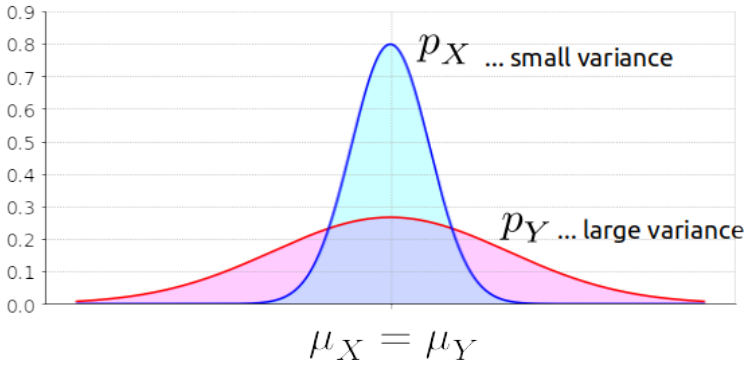

Figure 1.6: Small vs large variance

expanding the square and using properties 1 and 2 of the expected value.

$$\begin{aligned} Var[Y] &= E(Y - E[Y])^2 \\ &= E[Y^2 - 2E[Y]Y + (E[Y])^2] \\ &= E[Y^2] - 2E[Y]E[Y] + (E[Y])^2 \\ &= E[Y^2] - E[Y]^2 \end{aligned} \tag{1.34}$$

The variance of Y is therefore the difference between the mean of Y^2 an the squared mean of Y. The variance is a measure of the *spread* of a distribution (see Figure 1.6). When we sample a random variable with small variance, we can be fairly sure that the outcome will be close to the expected value. High-variance random variables on the other hand produce a wide range of outcomes. The variance measures the *uncertainty* captured by a random variable.

The unit of variance is the square of the unit of the outcome. For example, if the temperature is measured in $^\circ F$, then its variance has units $(^\circ F)^2$. For this reason we often report the square root of variance, known as the *standard deviation* of Y, and denoted with σ_Y .

In contrast with the expected value, the variance is *not* a linear function. Rather, the variance of a linear combination of random variables $\{Y_i\}_n$ is given by this more complicated nonlinear formula:

$$Var\left[\sum_{i=1}^{n} \alpha_i Y_i\right] = \sum_{i=1}^{N} \alpha_i^2 \, Var[Y_i] + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} \alpha_i \alpha_j Cov(Y_i, Y_j) \tag{1.35}$$

Notice here the two sources of nonlinearity. First the coefficients α_i in the first term on the right are squared. Second, the appearance of pair-wise *covariance* terms $Cov(Y_i, Y_j)$. We will define the covariance of a pair of random variables in part 1.9.

Example 1.5.1. Find the variance of the random variable of Example 1.3.1.

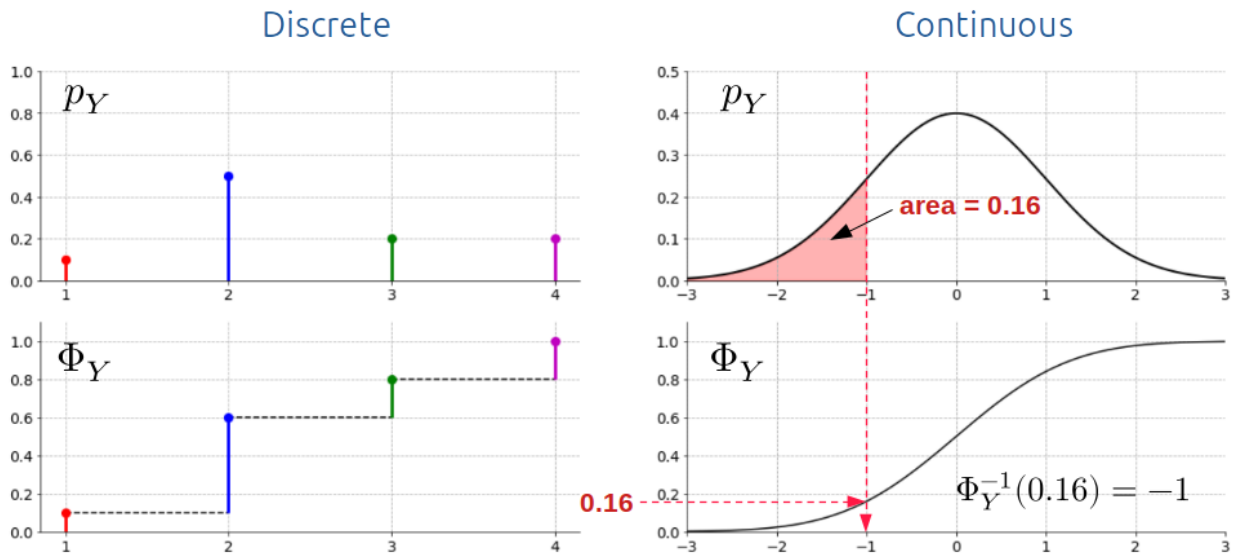

Figure 1.7: Discrete and continuous pdfs and cdfs. $\Phi_Y(y)$ is the probability of the event $(-\infty, y]$, which is obtained by integrating the pdf from $-\infty$ to y (inclusive). Integrating over the discrete pdf on the left produces a series of positive jump discontinuities. For the continuous distribution, the cdf is a continuous non-decreasing function. Its value at y equals the area under the pdf to the left of y.

Solution.

$$Var[X] = E\left[(X - E[X])^2\right] = E\left[\left(X - \frac{a}{a-1}\right)^2\right] \tag{1.36}$$

We apply Eq. 1.28 for the expected value of a function.

$$Var[X] = \int_1^\infty \left(x - \frac{a}{a-1}\right)^2 a x^{-1-a} dx \tag{1.37}$$

$$= a \int_1^\infty \left(x^2 - \frac{2a}{a-1}x + \frac{a^2}{(a-1)^2}\right) x^{-1-a} dx \tag{1.38}$$

$$= \vdots$$

$$= \frac{a}{(a-2)(a-1)^2} \qquad \text{provided } a > 2 \tag{1.39}$$

1.6 Cumulative distribution function

The *cumulative distribution function* (cdf) of a univariate random variable Y is a function from the sample space Ω_Y to the interval [0,1]. It is denoted with Φ_Y.

$$\Phi_Y : \Omega_Y \to [0, 1] \tag{1.40}$$

Evaluating Φ_Y on y returns the probability of obtaining a value less or equal to y.

$$\Phi_Y(y) = P(Y \leq y) \tag{1.41}$$

This can be calculated as the integral of the pdf from $-\infty$ to y.

$$\Phi_Y(y) = \int_{-\infty}^{y} p_Y(\xi)\, d\xi \tag{1.42}$$

As always, the integral should be interpreted as a sum when Y is discrete-valued, and in this case the sum must include $p_Y(y)$, since the definition uses the "$\leq$" symbol. Figure 1.7 shows discrete and continuous pdfs with their respective cdfs.

The cdf Φ_Y and the pdf p_Y are both equivalent to the probability measure P_Y, in the sense any of these serve to compute the probability of an event. The cdf is actually easier to use the the pdf, since it directly provides the probabilities of events, without the need for integration. Furthermore, it is a monotonic function, so it is invertible. The inverse of the cdf of Y, which we denote with Φ_Y^{-1} will feature prominently in parts 3.2 and 3.3 about confidence intervals and hypothesis tests.

1.7 Sampling and IID random variables

We "sample" a system when we take a measurement of the system. In our notation, the value of the measurement is y, and the system is represented by the random variable Y:

$$y \sim Y \tag{1.43}$$

But we usually do not take only a single measurement. A "sample" can more generally consist of N measurements:

$$\{y_i\}_N \sim Y \tag{1.44}$$

An alternative notation that will turn out to be more mathematically convenient, is to define a separate random variable for each of the samples in $\{y_i\}_N$: Y_1 for y_1, Y_2 for y_2, etc.. We can then write:

$$\{Y_i\}_N \overset{\text{iid}}{\sim} Y \tag{1.45}$$

Here we have defined a set of random variables $\{Y_1, \ldots, Y_N\}$ that are *independent and identically distributed* (iid). This is indicated with the $\overset{\text{iid}}{\sim}$ symbol. The Y_i's are "identical" in the sense that their individual (marginal) distributions are all equal to p_Y ($p_{Y_1} = p_Y$, $p_{Y_2} = p_Y$, etc.). The notion of "independence" relates to the interactions between multiple random variables and will be introduced in section 1.11. Briefly though, it means that a sample obtained from one of the Y_i's does not influence the samples of any other Y_j (for all i and j).

1.8 Law of Large Numbers

Consider the following. A friend proposes a game: Roll a die, if it comes up 1 or 2, you win one dollar; a 3, you win \$5; a 4 or 5, you win \$2; a 6, you loose \$12. Your "friend" assures you that you'll likely come out on top, because you win in 5 out of 6 plays. Should you play? Table 1.8 shows possible rewards r and their probabilities $p(r)$

r	\$-12	\$1	\$2	\$5
$p(r)$	$1/6$	$1/3$	$1/3$	$1/6$

Table 1.1: Outcomes and their probabilities

Your intuition is that the decision of whether or not to play should be based on the expected reward in each iteration of the game. So you begin by defining a random variable R for the reward, with $\Omega_R = \{-12, 1, 2, 5\}$, and pdf given in the table. Next, you compute the expected value of R.

$$E[R] = -12 \times 1/6 + 1 \times 1/3 + 2 \times 1/3 + 5 \times 1/6 = -1/6 \tag{1.46}$$

The expected reward is negative, and so you decide not to play. But wait! says your friend, why use the expected value? Why not the median? Or the *mode*? The mode is a third notion of average defined as the most common outcome; \$1 or \$2 in this case. Both of these are positive, your greedy friend insists.

The answer to this conundrum is given by the *law of large numbers*, which tells us that you are correct in using the expected value. It states that the average of N independent samples $\{y_1, \ldots, y_N\}$ of a random variable Y will converge to its expected value as N grows.

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} y_i = E[Y] \tag{1.47}$$

Figure 1.8 demonstrates the law of large numbers. The plot on the left shown a Gaussian distribution with zero mean ($\mu_Y = 0$) and standard deviation equal to 0.1 ($\sigma_Y = 0.1$). Each of the colored lines in the plot on the right show an evolution of $\frac{1}{N}\sum_{i=1}^{N} y_i$ as N increases. The law of large numbers predicts that all of these trajectories will converge to zero (the mean) as N increases.

1.9 Multivariate distributions

Most physical systems are not well described by a single measurement. We will now generalize our one-dimensional concepts of probability to multiple measurements; that is, to multi-dimensional sample spaces. We begin in two dimensions. Suppose I measure both the temperature and the humidity outside my window and represent these with random variables

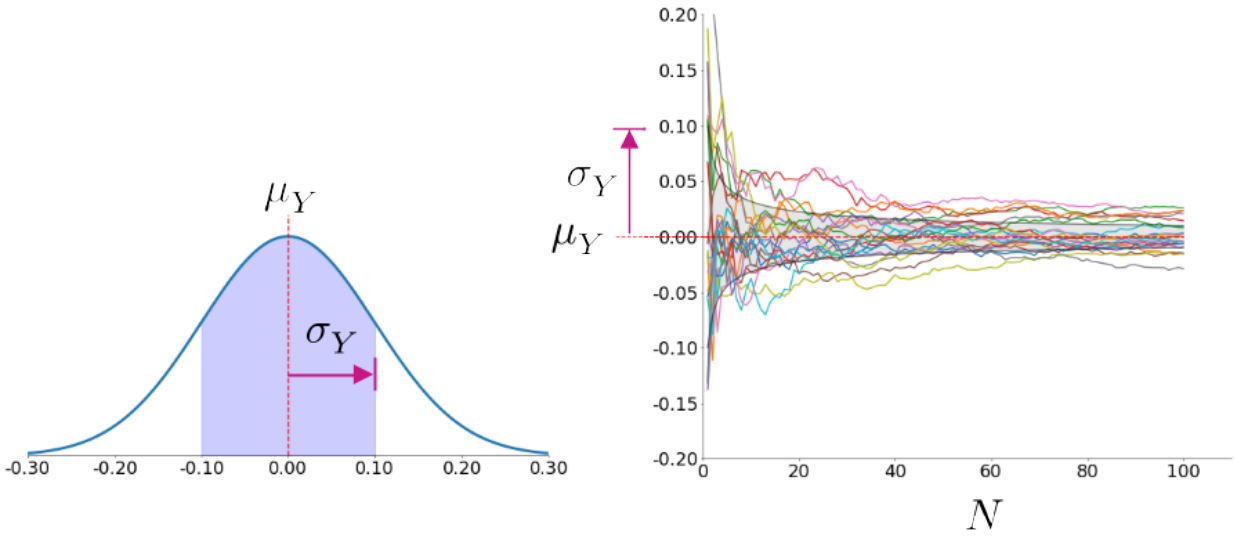

Figure 1.8: Each of the lines are averages of samples of size N taken from the distribution on the left. The law of large numbers asserts that these lines must converge to $\mu_Y = 0$ as N goes to ∞. (see `demo_lln.py`)

T and H. The *joint sample space* for T and H is the TH plane (i.e. $\mathbb{R}^2$), and an event is any region of that plane. For example, the event that the temperature is between 60°F and 65°F, while humidity is between 69% and 72%, is expressed as

$$T \in [60, 65] \text{ and } H \in [69, 72] \tag{1.48}$$

The probability measure P_{TH} assigns a value to each region in the sample space. For example,

$$P_{TH}(T \in [60, 65] \text{ and } H \in [69, 72]) = 0.02 \tag{1.49}$$

The *joint distribution* $p_{TH}(t, h)$ is the corresponding probability density function. It maps the sample space to the real numbers, and in this case can be visualized as a surface over the 2D plane (see Figure 1.9). As before, we can find the probability of an event by integrating the joint pdf over that event. In this case,

$$\int_{60}^{65} \int_{69}^{72} p_{TH}(t, h)\, dh\, dt = 0.02 \tag{1.50}$$

In general, we arrange any D random variables into a vector-valued *multivariate random variable*, with a corresponding multi-dimensional joint distribution.

$$Y = (Y^1, Y^2, \ldots, Y^D) \qquad \ldots \text{multivariate random variable} \tag{1.51}$$

$$p_Y : \mathbb{R}^D \to \mathbb{R} \qquad \ldots \text{multivariate distribution} \tag{1.52}$$

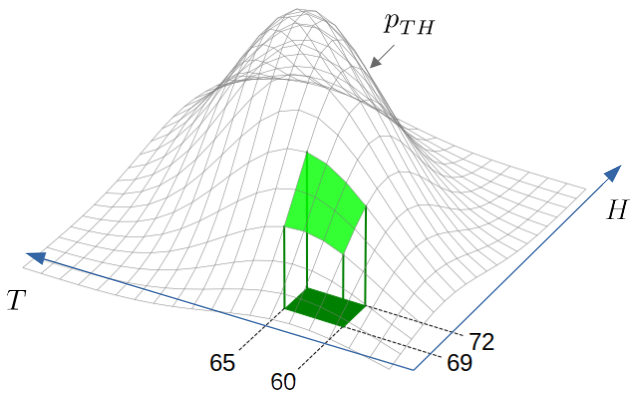

Figure 1.9: Joint distribution of temperature and humidity. The event that $T \in [60, 65]$ and $H \in [69, 72]$ is a rectangle in the horizontal TH plane. The probability of this event is the integral of p_{TH} over the rectangle.

Notice that we use superscripts to indicate a univariate component of a mulivariate random variable (e.g. Y^1), and subscripts to index variables in an iid collection (e.g. $\{Y_i\}_N$). The expected value of Y is a vector in $\mathbb{R}^D$, found by taking the expected value of each of its components:

$$E[Y] = (E[Y^1], E[Y^2], \ldots, E[Y^D]) \in \mathbb{R}^D \tag{1.53}$$

The variance of Y is a $D \times D$ matrix, defined as:

$$Var[Y] \;=\; E\left[(Y - E[Y])^T (Y - E[Y])\right] \;=\; \begin{bmatrix} \sigma_1^2 & \sigma_{1,2}^2 & \ldots & \sigma_{1,D}^2 \\ \sigma_{2,1}^2 & \sigma_2^2 & \ldots & \sigma_{2,D}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{D,1}^2 & \sigma_{D,2}^2 & \ldots & \sigma_D^2 \end{bmatrix} \tag{1.54}$$

We have assumed here that the multivariate Y is arranged as a *row* vector. This matrix is known as the *covariance matrix*. The i'th diagonal entry in the covariance matrix is the variance of Y^i (σ_i^2 in Eq. 1.54). The (i, j)'th entry of the covariance matrix is the *covariance* of Y^i and Y^j, defined as

$$Cov[Y^i, Y^j] = \sigma_{i,j}^2 = E[(Y^i - E[Y^i])(Y^j - E[Y^j])] \tag{1.55}$$

This is a scalar quantity. Notice that this definition is symmetric, in the sense that $\sigma_{i,j}^2 = \sigma_{j,i}^2$. The covariance matrix is therefore a symmetric matrix.

Example 1.9.1. Consider jointly distributed random variables X and Y, with sample space Ω_{XY} shown below (Ω_{XY} is the shaded triangle). Ω_{XY} is defined by three linear inequalities: $X \geq 0$, $Y \geq 0$, and $X + Y \leq 1$. The joint pdf of (X, Y) is,

$$p_{XY}(x, y) = \begin{cases} c\,(1 - x - y) & (x, y) \in \Omega_{XY} \\ 0 & \text{otherwise} \end{cases} \tag{1.56}$$

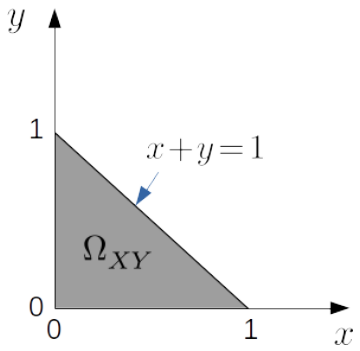

Figure 1.10: Sample space for Example 1.9.1

a) Compute c, b) Compute $P(X \leq 0.5)$

Solution. The distribution is an affine function of x and y, shown on the left side of Figure

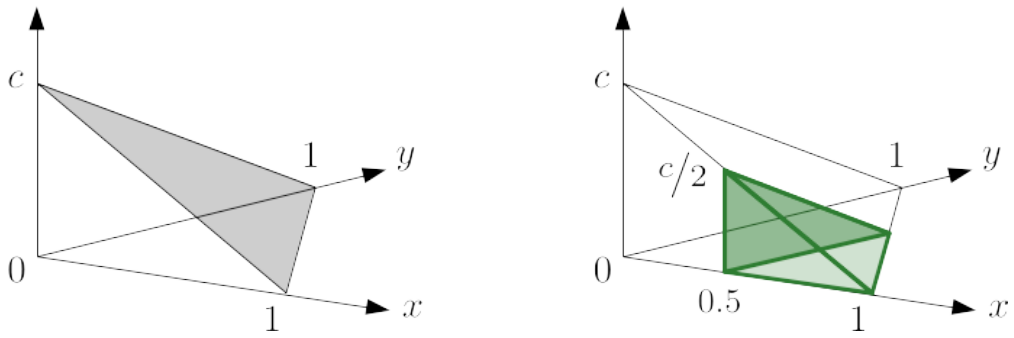

Figure 1.11: Equation 1.56

1.11. The integral is the volume of the triangular pyramid, which is one third its base times its height. This must equal 1.

$$1/3 \times \text{base} \times \text{height} = (1/3)\,(1/2)\,c = 1 \tag{1.57}$$

Therefore $c = 6$. Next, $P(X \leq 0.5)$ is the volume of the pyramid with the green portion shown on the right in Figure 1.11 removed.

$$\begin{aligned} P(X \leq 0.5) &= 1 - \text{volume of green shape} && (1.58)\\ &= 1 - 1/3 \times \text{base} \times \text{height} && (1.59)\\ &= 1 - 1/3 \times 1/8 \times c/2 \;=\; 7/8 && (1.60) \end{aligned}$$

New questions arise when we consider joint random variables: Are they predictive of one another, or do they vary independently? What can we learn about one random variable when we measure another? The concepts of marginal and conditional probabilities will help to address these questions.

1.9.1 Marginal distributions

Consider an D-dimensional random variable $Y = (Y^1, \ldots, Y^D)$. A *marginal* distribution can be constructed for any subset of component variables, and it represents the joint distribution of those variables when the others are ignored. Take for example the multivariate random variable of temperature and humidity, (T, H). The marginal distribution for T is obtained by integrating away (a.k.a. *marginalizing*) H:

$$p_T(t) = \int_{\Omega_H} p_{TH}(t, h) dh \tag{1.61}$$

Similarly, the marginal distribution of H is obtained by integrating the joint distribution over Ω_T:

$$p_H(h) = \int_{\Omega_T} p_{TH}(t, h) dt \tag{1.62}$$

Both of these formulas yield valid probability distributions, since (it can be shown) both $p_T(t)$ and $p_H(h)$ satisfy the requirements of positivity and unit integral. In general, we can compute the marginal distribution for any number of components $d < D$ simply by integrating away the other $D - d$ components.

Example 1.9.2. For Example 1.9.1, compute the marginal distributions of X and Y.

Solution. For each value of x, the marginal probability $p_X(x)$ is found by integrating over Ω_Y. We use the fact that $p_{XY}(x, y)$ is only non-zero between 0 and $1 - x$.

$$\begin{aligned} p_X(x) &= \int_{\Omega_Y} p_{XY}(x, y)\, dy && (1.63)\\ &= \int_0^{1-x} 6(1 - x - y)\, dy && (1.64)\\ &= \int_0^{1-x} 6(1 - x) dy - 6 \int_0^{1-x} dy && (1.65)\\ &= 6(1 - x)^2 - 6 \left.\frac{y^2}{2}\right|_0^{1-x} && (1.66)\\ &= 3(1 - x)^2 && (1.67) \end{aligned}$$

We can argue by symmetry that $p_Y(y) = 3(1 - y)^2$.

Example 1.9.3. For Example 1.9.1, compute $E[X]$.

Solution.

$$\begin{aligned} E[X] &= \int_{\Omega_X} x\, p_X(x) dx \\ &= \int_0^1 3x(1-x)^2 dx \\ &= 3\int_0^1 x(1-2x+x^2) dx \\ &= 3\left(\frac{1}{2}x^2 - \frac{2}{3}x^3 + \frac{1}{4}x^4\right)_0^1 \\ &= \frac{1}{4} \end{aligned}$$

Example 1.9.4. For Example 1.9.1, compute $Var[X]$.

Solution.

$$\begin{aligned} Var[X] &= E[(X - E[X])^2] \\ &= \int_{\Omega_X} (x - 1/4)^2\, p_X(x) dx \\ &= \int_0^1 (x - 1/4)^2\, 3(1-x)^2 dx \\ &= \vdots \\ &= 3/80 \end{aligned}$$

Note: Solving this integral by hand is tedious, however it can easily be done using the "sympy" package of Python, as shown below. There will not be any complicated integrals like this one on any tests in this course.

```
>> from sympy import symbols, integrate
>> x, y = symbols('x y')
>> p = 6*(1-x-y)
>> pX = integrate(p, (y,0,1-x))
>> EX = integrate(x*pX,(x,0,1))
>> VarX = integrate(((x-EX)**2)*pX, (x,0,1))
```

This code returns `VarX=3/80`

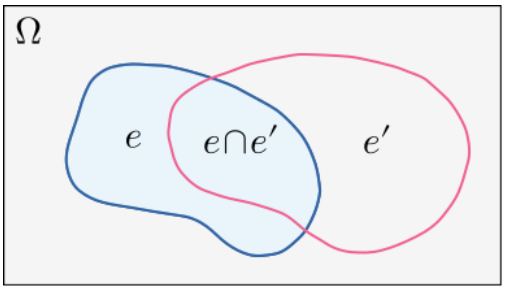

Figure 1.12: Dividing $P(e \cap e')$ by $P(e)$ amounts to rescaling the probability measure from a world where $P(\Omega) = 1$ to one where $P(e) = 1$.

1.9.2 Conditional probability

Conditional probability for events

To define the conditional probability, we must first introduce the notation for an event e' *conditioned* on another event e:

$$e' \mid e \tag{1.68}$$

This is pronounced "event e' given event e" or "event e' conditioned on event e". We define the probability of $e' \mid e$ with

$$P(e' \mid e) = \frac{P(e \cap e')}{P(e)} \tag{1.69}$$

That is, the probability of event e' given event e equals the probability that they both occur divided by the probability that event e occurs. The interpretation of $P(e' \mid e)$ is as the probability that event e' occurs when it is known that e occurs. This is illustrated in Figure 1.12.

Example 1.9.5. Find the probability that a 2 is obtained when rolling a six-sided die, assuming that the outcome is known to be less than 4.

Solution. Compute $P(\{2\} \mid \{1,2,3\})$ using the definition of conditional probability (Eq. 1.69):

$$P(\{2\} \mid \{1,2,3\}) = \frac{P(\{2\} \cap \{1,2,3\})}{P(\{1,2,3\})} = \frac{P(\{2\})}{P(\{1,2,3\})} = \frac{1/6}{1/2} = 1/3 \tag{1.70}$$

Conditional probability for univariate random variables

We just defined the probability that an *event* e' occurs, given that another event e occurs. We will now extend the concept to the entire *probability measure* of a random variable, given that event e occurs. We denote the random variable with $X = (\Omega_X, \mathcal{E}_X, P_X)$. Then we can define *a new random variable* $X|e$ (X conditioned on e, or X given e), whose probability

measure is computed with:

$$P_{X|e}(e') \;=\; P_X(e' \mid e) \;=\; \frac{P_X(e \cap e')}{P_X(e)} \qquad \forall e' \in \mathcal{E}_X \tag{1.71}$$

That is, we obtain the probability of e' under the conditioned measure $P_{X|e}$ by evaluating the conditional probability $e' \mid e$ using the unconditioned measure P_X, and this is given by Equation 1.69.

Conditional probability for multivariate random variables

In the previous example, the events e and e' were observations of a single quantity – the roll of a die. In the more general setting, the events may contain measurements of multiple quantities. That is, they may be subsets in a multi-dimensional sample space. Take for example the measurement of temperature and humidity. In this case, the conditioning event e might be the observation that temperature is between 60 and 65 degrees Fahrenheit while humidity is between 69 and 72 percent; a rectangle in Ω_{TH}.

$$e = T \in [60, 65] \text{ and } H \in [69, 72] \tag{1.72}$$

We can then construct a random variable $TH|e$ for the measurement of temperature and humidity conditioned on temperature being between 60 and 65 degrees Fahrenheit and humidity being between 69 and 72 percent. The probability measure for this random variable is, for an arbitrary event $e' \in \mathcal{E}_{TH}$:

$$P_{TH|e}(e') = \frac{P_{TH}(e \cap e')}{P_{TH}(e)} \tag{1.73}$$

What we've just seen is a very general definition of a conditional random variable, where the conditioning event e was allowed to by any event. We will now focus in on a special case that is common in practice. That is when the conditioning event e refers to a *measurement* of some or all of the components of the random variable. For example, e could be the event that temperature is known to be 62 degrees Fahrenheit. This is illustrated in Figure 1.13. Take an arbitrary event $\alpha \in \mathcal{E}_H$ and project it into Ω_{TH}. This procedure gives us an event e'. The conditional probability $H|T = 62$ is then a random variable with sample and event spaces Ω_H and $\mathcal{E}_H$, and probability measure given by:

$$P_{H|T=62}(e') = \frac{P_{TH}((T = 62) \cap e')}{P_T(T = 62)} \tag{1.74}$$

Even more generally, we may have a large number of quantities gathered into a multi-dimensional random variable Z. We now take measurements for a subset of those quantities, splitting Z into two parts $Z = (X, Y)$, where X are the measured quantities and Y are the unmeasured quantities. Measuring $X = x$ (x is an array of numbers) generates the

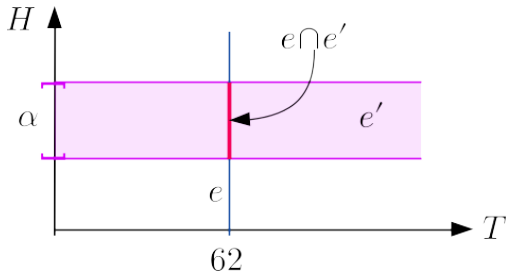

Figure 1.13: e is the vertical line through Ω_{TH} with $T = 62$.

conditional random variable $Y|X=x$ and for any $e' \in \Omega_Y$

$$P_{Y|X=x}(e') = \frac{P_Z((X = x) \cap e')}{P_X(X = x)} \tag{1.75}$$

It can be shown that the pdf corresponding to this random variable is:

$$p_{Y|X=x}(y) = \frac{p_Z(x, y)}{p_X(x)} \qquad \forall x \in \Omega_X, \forall y \in \Omega_Y \tag{1.76}$$

where $p_Z(x, y)$ is the pdf of the joint variable Z evaluated at (x, y). To prove Eq. 1.76, we must show that $p_{Y|X=x}(y)$ thus defined satisfies the axioms of probability, and that it corresponds to the probability measure of Eq. 1.75. A proof is provided in the box below, which is optional reading.

To prove that $p_{Y|X=x}(y)$ defined in Eq. 1.75 is a pdf, we must show that it is always positive, and that its integral over Ω_Y equals one. The first is easy: both the numerator and the denominator are positive (because they are pdfs), so their ratio is positive. For the second,

$$\int_{\Omega_Y} p_{Y|X=x}(y)dy = \int_{\Omega_Y} \frac{p_Z(x,y)}{p_X(x)}dy = \frac{\int_{\Omega_Y} p_Z(x,y)dy}{p_X(x)} = \frac{p_X(x)}{p_X(x)} = 1 \tag{1.77}$$

So $p_{Y|X=x}$ is indeed a valid pdf. To prove that it is the correct pdf, we must show that integrating it over an arbitrary $e' \in \mathcal{E}_Y$ yields $P_{Y|X=x}(e')$ as defined in Equation 1.75.

$$\begin{aligned}
\int_{e'} p_{Y|X=x}(y)dy &= \int_{e'} \frac{p_Z(x,y)}{p_X(x)}dy && (1.78)\\
&= \frac{\int_{e'} p_Z(x,y)dy}{p_X(x)} && (1.79)\\
&= \frac{\int_{e'} p_Z(x,y)dy}{\int_{\Omega_Y} p_Z(x,y)dy} && (1.80)\\
&= \frac{P((X=x)\cap e')}{P(X=x)} && (1.81)\\
&= P(e'|X=x) && (1.82)
\end{aligned}$$

Example 1.9.6. For Example 1.9.1, compute $p_{Y|X=1/2}(y)$.

Solution. We apply the formula for the pdf of a conditional random variable:

$$p_{Y|X=x}(y) = \frac{p_Z(x,y)}{p_X(x)} = \frac{6(1-x-y)}{3(1-x)^2} \tag{1.83}$$

With $x = 1/2$ this becomes:

$$p_{Y|X=1/2}(y) = \frac{6(1-1/2-y)}{3(1-1/2)^2} = 4(1-2y) \tag{1.84}$$

A note on notation

So far we have used our standard notation for probability density functions when referring to conditional random variables. That is, we've referred to the pdf of $Y|X=x$ evaluated at y as $p_{Y|X=x}(y)$. This is nice because it emphasizes the fact that a conditional random variable is itself a bona fide random variable. However the notation is a bit cumbersome, and not widely used. A more common and compact alternative is $p(Y=y \mid X=x)$ or even $p(y \mid x)$ for the pdf of $Y|X=x$ evaluated at y. We will use all three alternatives, as needed to balance readability and conciseness.

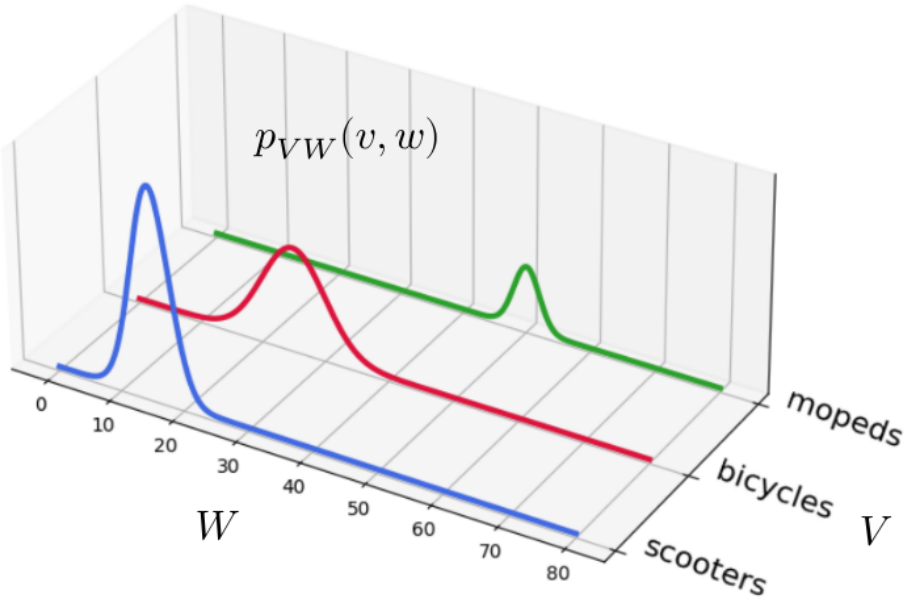

Figure 1.14: Joint distribution of weight and vehicle type

Example 1.9.7. A college campus has three types of vehicles: scooters, bicycles, and mopeds. The scooters are the lightest of the three, and also the most popular, accounting for 50% of the total. Bicycles are heavier than scooters and account for 40%, while the remaining 10% are mopeds and are the heaviest.

The joint distribution of vehicle type V and weight W is shown in Figure 1.14. Weight here is a continuous random variable with sample space $\Omega_W = \mathbb{R}$, and vehicle type is a categorical variable with $\Omega_V = \{\text{scooter, bicycle, moped}\}$. The joint distribution $p_{VW}(v, w)$ consists of three lines (if both were continuous, p_{VW} would be a surface). We can integrate the joint distribution over its sample space to confirm that it is a valid pdf.

$$\begin{aligned}\int_\Omega p_{VW}(v,w)dvdw &= \int_{\mathbb{R}} p_{VW}(\text{scooters}, w)dw + \int_{\mathbb{R}} p_{VW}(\text{bicycles}, w)dw + \int_{\mathbb{R}} p_{VW}(\text{mopeds}, w)dw \\ &= 0.5 + 0.4 + 0.1 \\ &= 1\end{aligned}$$

Figure 1.15 shows the marginal distributions of W and V, obtained by integrating the joint distribution over Ω_V and Ω_W, respectively. For $p_W(w)$, integrating over Ω_V amounts to summing over the three labels. For each $w \in \mathbb{R}$,

$$p_W(w) = \int_{\Omega_V} p_{VW}(v, w)dv \tag{1.85}$$

$$= p_{VW}(\text{scooter}, w) + p_{VW}(\text{bicycles}, w) + p_{VW}(\text{moped}, w) \tag{1.86}$$

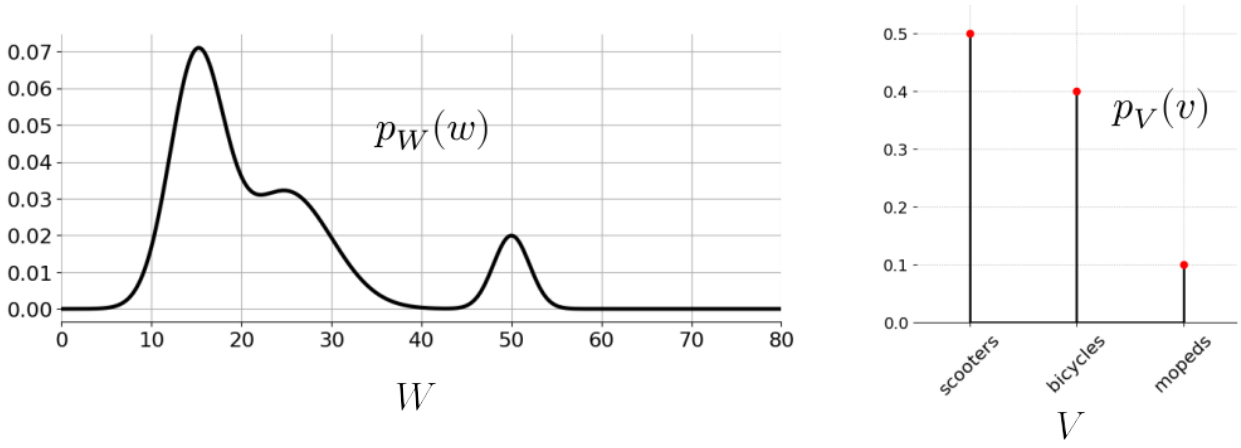

Figure 1.15: Marginal distributions

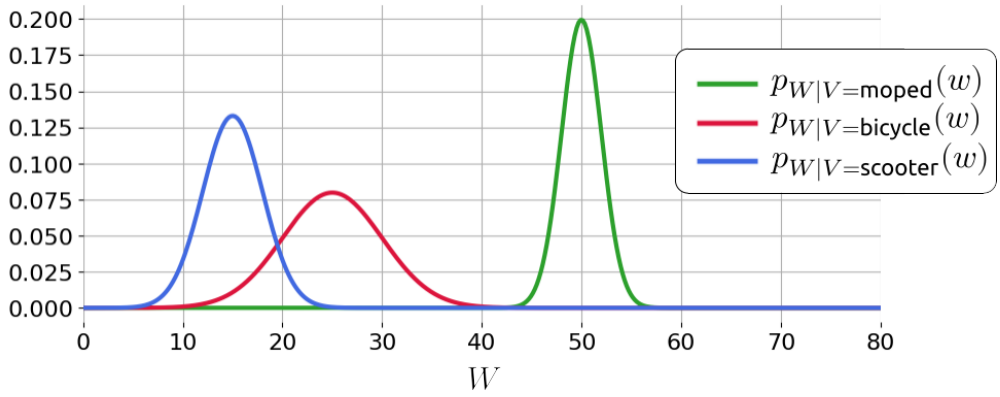

Figure 1.16: Distributions of weight conditioned on vehicle type.

p_V is obtained by integrating $p_{VW}(v, w)$ in the W direction.

$$p_V(\text{scooter}) = \int_{\mathbb{R}} p_{VW}(\text{scooter}, w)dw = 0.5 \tag{1.87}$$

$$p_V(\text{bicycle}) = \int_{\mathbb{R}} p_{VW}(\text{bicycle}, w)dw = 0.4 \tag{1.88}$$

$$p_V(\text{moped}) = \int_{\mathbb{R}} p_{VW}(\text{moped}, w)dw = 0.1 \tag{1.89}$$

Finally, the conditional probability of W given V is obtained by dividing the joint pdf by

the marginal of each vehicle class. This is shown in Figure 1.16. For each $w \in \mathbb{R}$,

$$p(w|\text{scooter}) = \frac{p_{VW}(\text{scooter}, w)}{p_V(\text{scooter})} = \frac{p_{VW}(\text{bicycle}, w)}{0.5} \tag{1.90}$$

$$p(w|\text{bicycle}) = \frac{p_{VW}(\text{bicycle}, w)}{p_V(\text{bicycle})} = \frac{p_{VW}(\text{scooter}, w)}{0.4} \tag{1.91}$$

$$p(w|\text{moped}) = \frac{p_{VW}(\text{moped}, w)}{p_V(\text{moped})} = \frac{p_{VW}(\text{moped}, w)}{0.1} \tag{1.92}$$

Example 1.9.8. For the distribution of Example 1.9.1, find the pdf of $Y|X = 0.5$.

Solution. Apply Eq. 1.76.

$$p(Y=y \mid X=0.5) = \frac{p_{XY}(0.5, y)}{p_X(0.5)} \tag{1.93}$$

$$= \frac{6(1-y-0.5)}{3(1-0.5)^2} \tag{1.94}$$

$$= 4 - 8y \tag{1.95}$$

1.10 Bayes' rule

Using equation 1.69, we can calculate the probability that two events A and B occur, as the product of the conditional probability of B given A and the probability of A:

$$P(A \cap B) = P(B|A)P(A) \tag{1.96}$$

We can also use this equation to calculate $P(A \cap B)$ in terms of $P(A|B)$ and $P(B)$:

$$P(A \cap B) = P(A|B)P(B) \tag{1.97}$$

Combining these two produces **Bayes' rule**, also known as Bayes' theorem:

$$P(B|A) = \frac{P(A|B)}{P(A)} P(B) \tag{1.98}$$

As with the conditional probability, there is also a version of Bayes' rule for pdfs:

$$p(X=x|Y=y) = \frac{p(Y=y|X=x)}{p(Y=y)} p(X=x) \tag{1.99}$$

Despite its simplicity, Bayes' rule has many useful applications. If we adopt the Bayesian interpretation of probabilities, in which we think of the events A and B as two hypotheses about the world, and their probabilities $P(A)$ and $P(B)$ as our degrees of belief in those hypotheses, then Eq 1.98 expresses how our belief in hypothesis B is updated when we

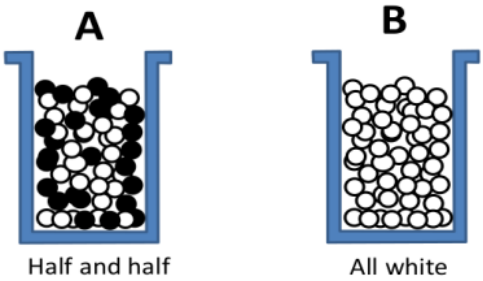

Figure 1.17: Two urns with black and white marbles.

observe A to be true.

Example 1.10.1. As an example, let's take B to represent the event that my car will start the next time I turn the ignition. Whether or not it starts will depend on many factors, including the amount of gas in the tank. We will denote with A the event that there is sufficient gas in the tank to start the car. My car is fairly old, and it only starts about 90% of the time. So my "prior belief" that it will start is $P(B) = 0.9$. Furthermore, I tend to be somewhat forgetful about filling the tank. There is about a 5% chance that the tank is empty, so my prior belief about event A is $P(A) = 0.95$. $P(A|B)$ is the probability that the car has gas if it has been observed to start. This equals 1 since a car with no gas can never start. Bayes' rule can then be used to update my belief that the can will start once I have observed that there is gas in the tank:

$$P(B|A) = \frac{1}{0.95}0.9 = 0.947 \tag{1.100}$$

Upon observing that the car has gas, my belief that it will start increases from a prior value of 0.9 to a posterior value of 0.947.

Example 1.10.2. One of the two urns shown in Figure 1.17 is placed in front of you, but you do not know which. You are asked to pick a marble. Before looking at the marble, what is your degree of belief for the proposition that you've picked from urn A? Urn B? How do these degrees of belief change if the marble turns out to be white? Black?
Solution.
The a-priori beliefs for urns A and B are both 0.5.

$$P(A) = P(B) = 0.5 \tag{1.101}$$

The fact that A has half white and half black marbles, while B has all white is captured with conditional probabilities.

$$P(\text{white}|A) = P(\text{black}|A) = 0.5 \tag{1.102}$$
$$P(\text{white}|B) = 1 \tag{1.103}$$
$$P(\text{black}|B) = 0 \tag{1.104}$$

We use Bayes' rule to compute the reverse conditional probabilities.

$$P(A|\text{white}) = \frac{P(\text{white}|A)P(A)}{P(\text{white})} \tag{1.105}$$

$$= \frac{P(\text{white}|A)P(A)}{P(\text{white}|A)P(A) + P(\text{white}|B)P(B)} \tag{1.106}$$

$$= \frac{0.5 \times 0.5}{0.5 \times 0.5 + 1 \times 0.5} \tag{1.107}$$

$$= 1/3 \tag{1.108}$$

$P(B|\text{white})$ is therefore $2/3$. We can repeat the computation for the case that we picked a black marble.

$$P(A|\text{black}) = \frac{P(\text{black}|A)P(A)}{P(\text{black})} \tag{1.109}$$

$$= \frac{P(\text{black}|A)P(A)}{P(\text{black}|A)P(A) + P(\text{black}|B)P(B)} \tag{1.110}$$

$$= \frac{0.5 \times 0.5}{0.5 \times 0.5 + 0 \times 0.5} \tag{1.111}$$

$$= 1 \tag{1.112}$$

And therefore $P(B|\text{black}) = 0$.

A common usage of Bayes' rule is in anti-causal inference. If in Eq. 1.99 X represents a cause and Y its effect, then we may, through experimentation, gather data for the conditional distribution $p(Y = y \mid X = x)$ by applying different inputs x to a system and observing the resulting y. Bayes' rule can then be used to compute $p(X = x \mid Y = y)$: the probability of that the cause X had value x, given that we have observed result $Y = y$.

Example 1.10.3. In a previous example we computed the probability of the vehicle's weight conditioned on the vehicle's type. It is reasonable to think of the vehicle type as "causing" its weight, and not the other way around. Bayes' rule can then be used to infer the probability of vehicle's type given its weight:

$$p(V = v \mid W = w) = \frac{p(W = w \mid V = v)}{p_W(w)} p_V(v) \tag{1.113}$$

Figure 1.18 shows the result of this computation for each of the three vehicle types, and expressed as a function of weight. Notice that, for each weight, the sum of the three lines equals 1. The plot implies the following rule for guessing the type of a vehicle based on its weight:

- $W < 20$, guess a scooter.
- $W \in [20, 42]$, guess a bicycle.

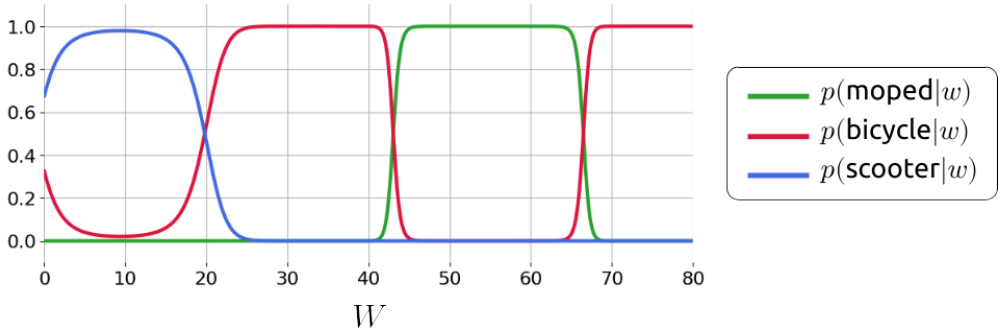

Figure 1.18: Vehicle type conditioned on the weight, as a function of the weight.

- $W \in [42, 66]$, guess a moped.
- $W > 66$, guess a bicycle.

It is surprising that, even though mopeds are on average heavier than bicycles, we should expect that very heavy vehicles ($W > 66$) are bicycles. This is due to the fact that the variance in the weight of bicycles is larger than that of mopeds. If the three conditional random variables $W \mid V = \text{bicycle}$, $W \mid V = \text{scooter}$, and $W \mid V = \text{mopeds}$ had equal variance, then this sort of reversal would not occur. We will return to this topic when we study logistic regression.

1.11 Independence

In the examples of the previous section, the act of obtaining some information about one event or a random variable, changed our belief about another event or random variable. Observing that there was gas in the tank increased my belief that my car would start; drawing a black marble from an urn decreased my belief that it contained only white marbles; observing that a vehicle has a low weight increases our belief that it is a bicycle. These events are *not* independent. On the other hand, the fact that the day is hot does not influence our belief about the roll of a die. Weather and dice are independent random variables.

The mathematical definition of independence states that two events e and e' are *independent* if the probability of e' is unaffected by observing e.

$$P(e' \mid e) = P(e') \tag{1.114}$$

This is equivalent (by Bayes' rule) to stating that the prior probability of e is unaffected by observing e'.

$$P(e \mid e') = P(e) \tag{1.115}$$

Independence is therefore a pair-wise relation: if e is independent of e', then e' is also independent of e, and (e, e') are said to be an independent pair of events. A third equivalent

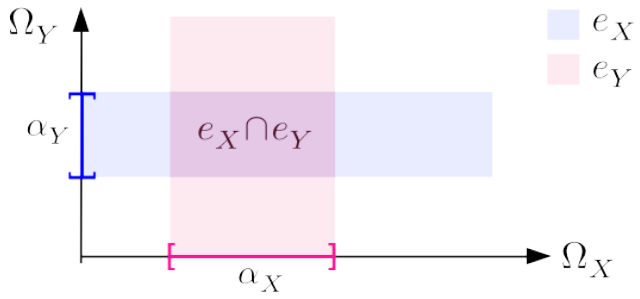

Figure 1.19: Independent random variables.

definition is that the probability of observing both e and e' equals the product of the probabilities of observing each:

$$P(e \cap e') = P(e)P(e') \tag{1.116}$$

These definitions of independence of events can be extended to a definition for random variables, much as we did with conditional probability and Bayes' rule. Roughly speaking, two random variables X and Y are independent if no event in one can inform any event in the other. That is, the event-based definition of independence must hold for every pair of events in the event spaces of X and Y. This is illustrated in Figure 1.19 where α_X and α_Y are arbitrary events in $\mathcal{E}_X$ and $\mathcal{E}_Y$ respectively. The condition for independence of random variables X and Y is then:

$$P_{XY}(e_X \cap e_Y) = P_X(\alpha_X)\, P_Y(\alpha_Y) \qquad \forall\, \alpha_X \in \mathcal{E}_X, \alpha_Y \in \mathcal{E}_Y \tag{1.117}$$

Here e_X and e_Y are respectively the projections of α_X and α_Y into the joint sample space Ω_{XY}. This can be expressed more succinctly as a condition on the joint and marginal distributions of X and Y:

$$p_{XY}(x, y) = p_X(x)\, p_Y(y) \tag{1.118}$$

Proof: In Eq. 1.117, substitute the probability measures for their expressions as integrals of pdfs:

$$\int_{e_X \cap e_Y} p_{XY}(x, y)\, dx\, dy = \left(\int_{\alpha_X} p_X(x)\, dx\right)\left(\int_{\alpha_Y} p_Y(y)\, dy\right) \tag{1.119}$$

Both sides of this equation can be written as a double integrals over x and y:

$$\int_{\alpha_Y}\int_{\alpha_X} p_{XY}(x, y)\, dx\, dy = \int_{\alpha_Y}\int_{\alpha_X} p_Y(y) p_X(x)\, dx\, dy \tag{1.120}$$

Since the intervals were arbitrary, this implies that the integrands must be equal to each other:

$$p_{XY}(x, y) = p_Y(y)\, p_X(x) \tag{1.121}$$

□

The analogs of Eqs. 1.114 and 1.115 for random variables are:

$$p(Y=y \mid X=x) = p_Y(y) \tag{1.122}$$
$$p(X=x \mid Y=y) = p_X(x) \tag{1.123}$$

or more compactly:

$$p(y|x) = p(y) \tag{1.124}$$
$$p(x|y) = p(x) \tag{1.125}$$

Proof
We will only prove Eq. 1.124 only, Eq. 1.125 being symmetric. The result is an immediate consequence of the definition of condition probability (Eq. 1.76).

$$p(Y=y \mid X=x) \;=\; \frac{p_{XY}(x,y)}{p_X(x)} \;=\; \frac{p_X(x)\,p_Y(y)}{p_X(x)} \;=\; p_Y(y) \tag{1.126}$$

□

The covariance of two independent random variables is zero:

$$X \text{ and } Y \text{ are independent} \quad \Rightarrow \quad Cov(X,Y) = 0 \tag{1.127}$$

Proof: Start with the expected value of the product of X and Y:

$$E[XY] = \int_{\Omega_{XY}} p_{XY}(x,y)\,dx\,dy \tag{1.128}$$

Since X and Y are independent, we can apply Eq. 1.121:

$$E[XY] = \int_{\Omega_{XY}} p_Y(y)\,p_X(x)\,dx\,dy \tag{1.129}$$

The double integral can be separated into two simple integrals:

$$E[XY] = \int_{\Omega_X} p_X(x)\,dx \int_{\Omega_Y} p_Y(y)\,dy = E[X]E[Y] \tag{1.130}$$

This is actually an interesting finding: that the expected value of a product of independent random variables is the product of their expected values. Next we define $\mu_X = E[X]$ and $\mu_Y = E[Y]$, and note the following identity, obtained using the linearity of the expected value (Eq. 1.27):

$$2\mu_X\mu_Y = E[\mu_X Y + \mu_Y X] \tag{1.131}$$

Combining equations 1.130 and 1.131:

$$E[XY] + 2\mu_X\mu_Y = \mu_X\mu_Y + E[\mu_X Y + \mu_Y X] \tag{1.132}$$

Rearranging and again using the linearity of the expectation we get:

$$E[XY + \mu_X\mu_Y - \mu_X Y - \mu_Y X] = 0 \tag{1.133}$$

which under further rearrangement becomes the covariance:

$$E[(X - \mu_X)(Y - \mu_Y)] = 0 \tag{1.134}$$

□

The concept of independence can be extended to more than two random variables. A general multivariate random variable $(Y_1, \ldots, Y_n)$ is *pairwise independent* if every pair of its components is independent in the sense of Eq. 1.121.

$$p_{Y_iY_j}(y_i, y_j) = p_{Y_i}(y_i)p_{Y_j}(y_j) \tag{1.135}$$

A stronger notion of multivariate independence called *mutual independence* requires that the distribution factor completely into a product of univariate distributions:

$$p_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n) = \prod_{i=1}^{n} p_{Y_i}(y_i) \tag{1.136}$$

Mutual independence is stronger than pair-wise independence: a collection of random variables might be pair-wise independent and *not* mutually independent. In this course we will always take "independence" of a set of random variables to mean mutual independence.

1.12 Correlation

Correlation is a more fine-grained concept than independence. It tells us the degree to which two random variables follow linear tendency. The correlation between random variables X and Y is quantified with the *correlation coefficient* ρ_{XY}.

$$\rho_{XY} = \frac{Cov(X, Y)}{\sigma_X\sigma_Y} \tag{1.137}$$

Equation 1.127 shows us immediately that independent variables have zero covariance (they are "uncorrelated"). However the opposite is not true: uncorrelated variables need not be independent. To gain some intuition on the correlation coefficient, notice that it can be rewritten as follows:

$$\rho_{XY} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X\sigma_Y} \tag{1.138}$$

$$= E\left[\left(\frac{X - \mu_X}{\sigma_X}\right)\left(\frac{Y - \mu_Y}{\sigma_Y}\right)\right] \tag{1.139}$$

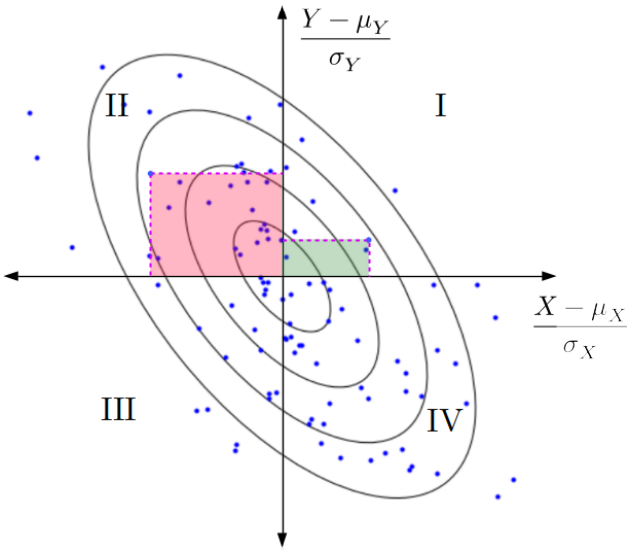

Figure 1.20: Scatter plot of the normalized measurements.

In this equation, $\frac{X-\mu_X}{\sigma_X}$ and $\frac{Y-\mu_Y}{\sigma_Y}$ are "normalized" versions of X and Y. They are normalized in that they have been shifted to have zero mean (by subtracting the mean) and scaled to have unit variance (by dividing by the standard deviation).

Let's look at some data. Figure 1.20 shows a scatter plot of data sampled from (X, Y) and normalized to zero mean / unit variance. The expectation from Eq. 1.139 can be approximated as the average of the product of the x and y coordinates of points in the figure. For samples in quadrants `I` and `III`, this product is positive (e.g. area of the green box), and for samples in quadrants `II` and `IV` the product is negative (e.g. negative area of the red box). The average is then positive if the samples tend to fall into quadrants `I` and `III`, negative if they tend to fall into quadrants `II` and `IV`, and zero if they fall equally into both pairs of quadrants.

Because the correlation is an average of normalized samples, its value is in the range [-1, 1]. The extreme values of $\rho_{XY} = -1$ and $\rho_{XY} = 1$ correspond to perfect linear dependence between the two variables. They obey a linear equation of the form $Y = \alpha X$, so knowing the value of one precisely determines the value of the other. $\rho_{XY} = 0$ means that X and Y are *uncorrelated*. We've already noted that this does not imply that they are independent, only that they are similarly distributed between quadrants (`I`, `III`) and (`II`, `IV`).

Figure 1.21 provides several examples. Each subplot shows a scatter plot of data sampled from a joint distribution p_{XY}. In the top row, X and Y are jointly Gaussian (Gaussian distributions are described in section 1.13.6). In the middle plot of the top row, the two variables are uncorrelated ($\rho_{XY} = 0$). In the special case of Gaussian variables, uncorrelatedness *does* imply independence. The middle row shows several cases of perfect correlation: $Y = \alpha X$. Here, $\rho_{XY} = -1$ when $\alpha < 0$, and $\rho_{XY} = 1$ when $\alpha > 0$. When $\alpha = 0$ (middle plot), ρ_{XY} is undefined. The bottom row exhibits cases of uncorrelated variables that are nevertheless not independent. Knowing the value of one *does* provide information about the

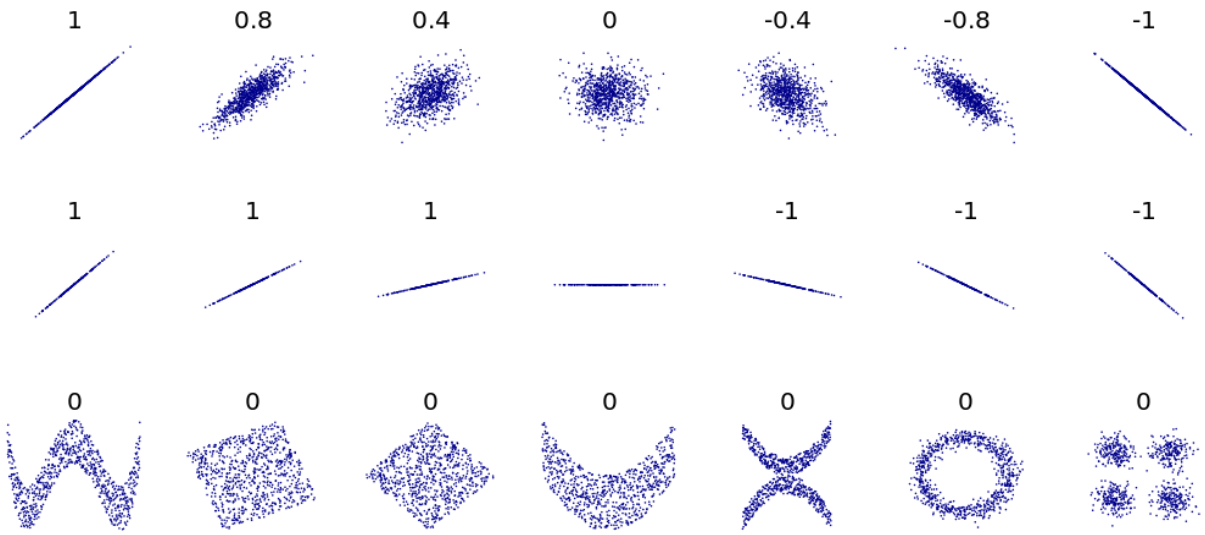

Figure 1.21: Correlation coefficient

other. However this relation is not monotonic: larger values of X do not imply larger (or smaller) values of Y.

Example 1.12.1. Find the correlation coefficient for the distribution of Example 1.9.1.

Solution. The standard deviations were found in Example 1.9.4 to be $\sigma_X = \sigma_Y = \sqrt{3/80}$. We can plug these into Eq. 1.139 and use the definition of the expectation to obtain the answer. The integral is too tedious to do by hand, and so we resort to sympy.

```
>> from sympy import symbols,integrate,sqrt
>> x, y = symbols('x y')
>> p = 6*(1-x-y)
>> pX = integrate(p,(y,0,1-x))
>> pY = integrate(p,(x,0,1-y))
>> EX = integrate(x*pX,(x,0,1))
>> EY = integrate(y*pY,(y,0,1))
>> VarX = integrate(((x-EX)**2)*pX,(x,0,1))
>> VarY = integrate(((y-EY)**2)*pY,(y,0,1))
>> rhoXY = integrate((x-EX)*(y-EY)/sqrt(VarX*VarY)*p, (y,0,1-x), (x,0,1))
```

This code returns $\rho_{XY} = -1/3$

1.13 Parametric pdfs

Here we introduce a few important families of pdfs with proper names. You can find a long list of such distributions in the article titled "List of probability distributions" in Wikipedia.

These are "parametric" families, in the sense that they are expressed with a mathematical formula with some number of tunable parameters. A "member" of a family is a particular distribution obtained by setting the parameters to given values. We denote this with $p(y; \theta_1, \theta_2, \ldots, \theta_P)$. Here θ_i is a value assigned to the i'th parameter of a family with P parameters. The semi-colon in the notation separates the arguments of the pdf (values in the sample space) from its parameters.

1.13.1 Bernoulli distribution $\mathcal{B}(p)$

The Bernoulli distribution is the simplest model for a discrete-valued random variable. Its sample space consists of just two outcomes. Call them "success" and "failure", and denote them with ✓ and ✗.

$$\Omega = \{✗, ✓\} \tag{1.140}$$

The Bernoulli distribution has only one parameter which is the probability p of success.

$$p(k; p) = \begin{cases} p & k = ✓ \\ 1 - p & k = ✗ \end{cases} \tag{1.141}$$

We use symbol $\mathcal{B}$ for the Bernoulli family, and $\mathcal{B}(p)$ for the particular distribution with parameter value p. $Y \sim \mathcal{B}(p)$ means that the pdf of Y is the Bernoulli distribution with parameter value p.

To do computations with Bernoulli random variables, we need to assign numbers to outcomes ✓ and ✗. There are two commonly used alternatives: $\Omega = \{0, 1\}$ and $\Omega = \{-1, 1\}$. Which one to use is purely a matter of mathematical convenience.

$\{0, 1\}$ encoding:

$$p(k; p) = \begin{cases} p & k = 1 \\ 1 - p & k = 0 \end{cases} \tag{1.142}$$

$\{-1, 1\}$ encoding:

$$p(k; p) = \begin{cases} p & y = 1 \\ 1 - p & y = -1 \end{cases} \tag{1.143}$$

For the purpose of taking derivatives, it is often convenient to define a smooth extension of $p(k; p)$. This is a function that passes through the two points of the discrete distribution and also has a continuous first derivative. Figure 1.22 shows two options: linear and exponential. Below are the formulas for both, in each of the two encodings.

$\{0, 1\}$ encoding:

$$p(k; p) = p^k (1 - p)^{1-k} \qquad \text{exponential} \tag{1.144}$$

$$p(k; p) = pk + (1 - p)(1 - k) \qquad \text{linear} \tag{1.145}$$

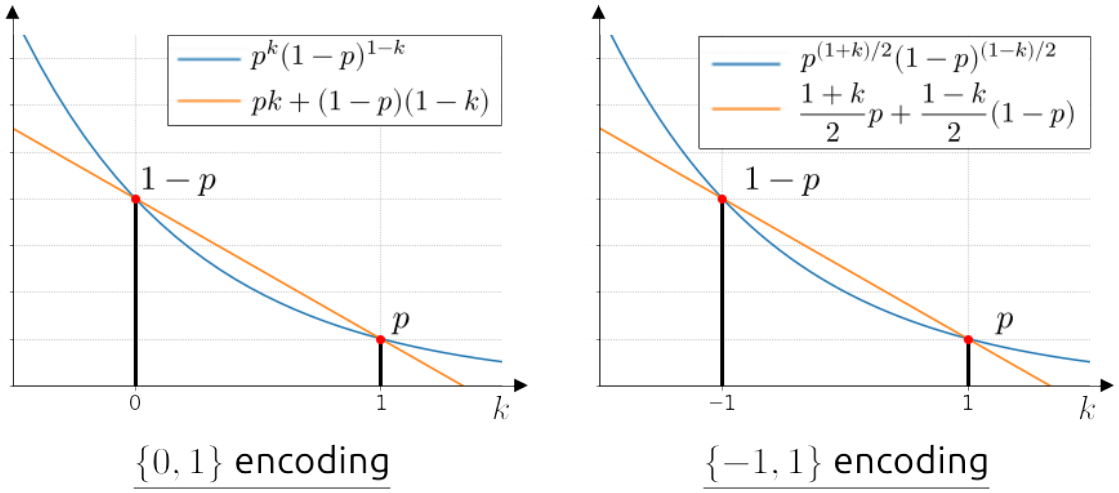

Figure 1.22: Smooth extensions of the Bernoulli pdf, in each of two numerical encodings.

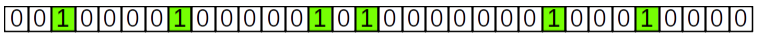

Figure 1.23: A sample sequence of Bernoulli trials with $p = 0.25$.

$\{-1,1\}$ encoding:

$$p(k;p) = p^{(1+k)/2}(1-p)^{(1-k)/2} \qquad \text{exponential} \qquad (1.146)$$

$$p(k;p) = \frac{1+k}{2}p + \frac{1-k}{2}(1-p) \qquad \text{linear} \qquad (1.147)$$

1.13.2 Binomial distribution $\mathcal{B}in(N,p)$

The binomial distribution applies to the total number of successes in an independent set of N Bernoulli trials $\mathcal{B}(p)$. For example, it applies to the number k of people with a particular illness in a random sample of N people, when each person has a probability p of having the disease. The sample space for the binomial distribution is $\Omega = \{0, \ldots, N\}$, and its pdf is:

$$p(k;N,p) = \binom{N}{k} p^k (1-p)^{N-k} \qquad (1.148)$$

Here $k \in \Omega$, N is the total number of trials, and p is the probability of success in each trial. Notice how this reduces to Eq. 1.144 when $N = 1$ (use $0! = 1$). Figure 1.23 provides an illustration. It shows an outcome for a sequence of 32 Bernoulli trials with $p = 0.25$. Of the 32 trials, 6 are successes. This is fewer than expected. The expectation of a binomial distribution is Np. That is, $Y \sim \mathcal{B}in(N,p)$, $E[Y] = Np$. The probability of the sequence in

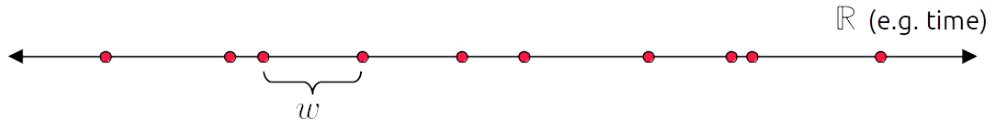

Figure 1.24: A Poisson process.

Figure 1.23 is obtained by evaluating the pdf of $\mathcal{B}in(32, 0.25)$ at $k = 6$:

$$p(6\,;N{=}32,p{=}0.25) = \binom{32}{6} (1/4)^6 \, (3/4)^{26} \approx 9.06 \times 10^5 \frac{2.54 \times 10^{12}}{1.84 \times 10^{19}} \approx 0.125 \tag{1.149}$$

1.13.3 Poisson process

A *stochastic process* is a mathematical object for representing that generate events over time. This is a similar concept as the sequence of events of Figure 1.23, except that now they occur on the real line (the axis of time), as shown in Figure 1.24. Stochastic processes are an interesting and large topic, and we only touch upon briefly in the chapter about time series data. The Poisson process is a very simple process that assumes only that all events are independent, and that the expected number of events in any two equally-sized intervals is the same. This means that there is a positive number λ such that the expected number of events in an interval of length Δt is $\lambda \Delta t$. We call this number the *rate* of the process, and it is measured in events per unit of time.

The *Poisson random variable* or *Poisson distribution* counts the number of events in a Poisson process during one unit of time. The expected value of a Poisson random variable is clearly λ, by definition. Its sample space is all of the natural numbers, including 0:

$$\Omega = \mathbb{N}_0 \tag{1.150}$$

Its pdf is

$$p(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1.151}$$

This formula can be obtained as the limit of a binomial distribution when the number N of trials in one unit of time goes to infinity.

Proof. We imagine the Bernoulli trials of Figure 1.23 as occurring in time, one after the other, over a period of one time unit (one second, for example). Define λ as the expected

number of successes in that period: $\lambda = pN$. Then, applying Eq. 1.148,

$$p(k; N, p) = \binom{N}{k} p^k (1-p)^{N-k} \tag{1.152}$$

$$= \frac{N!}{k!(N-k)!} \left(\frac{\lambda}{N}\right)^k \left(1 - \frac{\lambda}{N}\right)^{N-k} \tag{1.153}$$

$$= \frac{N(N-1)\dots(N-k+1)}{k!} \frac{\lambda^k}{N^k} \left(1 - \frac{\lambda}{N}\right)^{N-k} \tag{1.154}$$

$$= \frac{\lambda^k}{k!} \frac{N(N-1)\dots(N-k+1)}{\underbrace{N \times N \times \dots \times N}_{K \text{times}}} \left(1 - \frac{\lambda}{N}\right)^{N-k} \tag{1.155}$$

Take the limit as $N \to \infty$.

$$\lim_{N\to\infty} p(k; N, p) = \frac{\lambda^k}{k!} \lim_{N\to\infty} \left(1 - \frac{1}{N}\right) \dots \left(1 - \frac{k-1}{N}\right) \left(1 - \frac{\lambda}{N}\right)^{-k} \left(1 - \frac{\lambda}{N}\right)^{N} \tag{1.156}$$

Notice that all of the terms except the last tend to 1 as $N \to \infty$. So we can remove them from the limit. This limit of the last term is a common identity in calculus which we will not derive here. It equals $e^{-\lambda}$. Hence,

$$\lim_{N\to\infty} p(k; N, p) = \frac{\lambda^k}{k!} e^{-\lambda} \tag{1.157}$$

1.13.4 Exponential distribution $\mathcal{E}(\lambda)$

The exponential distribution models the waiting time between events in a Poisson process. A sample waiting time is shown as w in Figure 1.24. This is a continuous-valued distribution. Its sample space is the positive real numbers: $\Omega = \mathbb{R}^+$. The pdf for the waiting times is:

$$p(w; \lambda) = \lambda e^{-\lambda w} \tag{1.158}$$

1.13.5 Uniform distribution $\mathcal{U}(a, b)$

A uniform distribution is one whose sample space is an interval ()$\Omega = [a, b]$), and whose pdf is a constant C on that interval.

$$p(y; a, b) = \begin{cases} C & \text{if } y \in [a, b] \\ 0 & \text{otherwise} \end{cases} \tag{1.159}$$

Figure 1.25 shows the two types of uniform distribution: continuous and discrete. The value of C is a function of the two parameters of the distribution (a and b). Saying that something is "uniformly distributed" means that all outcomes in the interval $[a, b]$ are equally probably, and no other values are possible. Examples of discrete uniform random quantities abound:

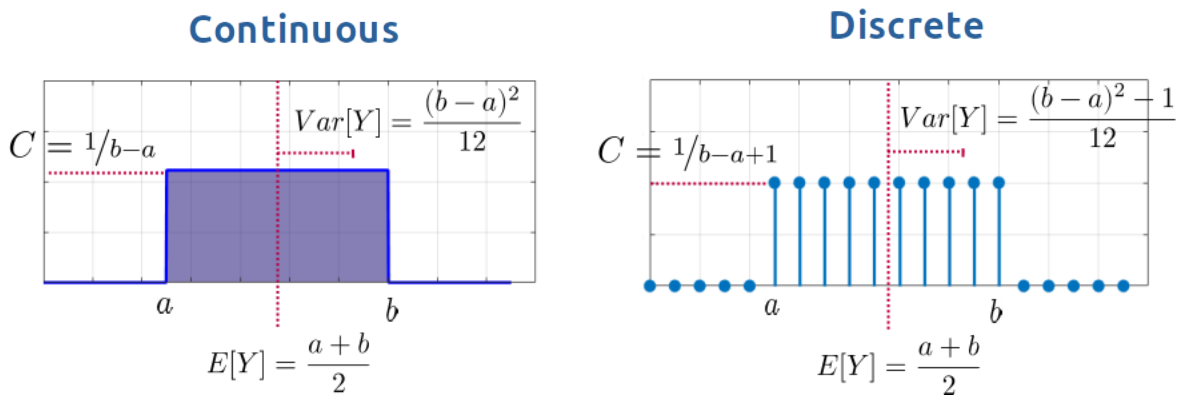

Figure 1.25: Continuous and discrete uniform distributions

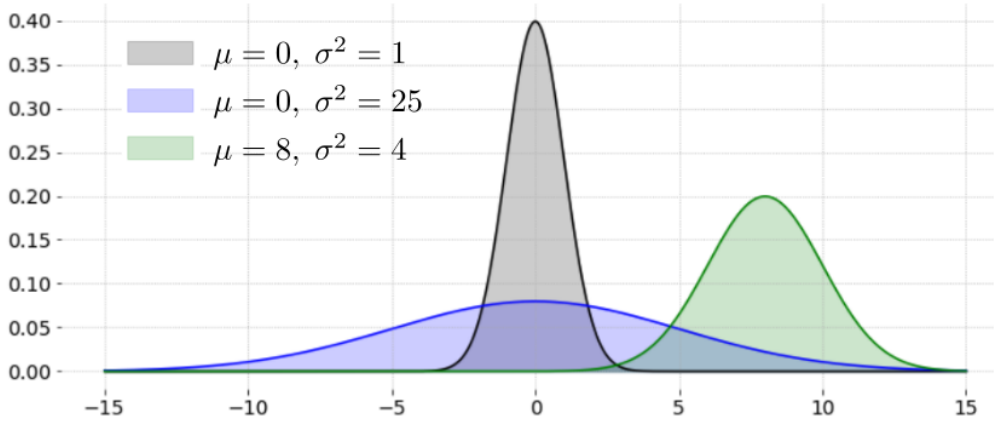

Figure 1.26: Normal probability density functions

rolling a single die, flipping a fair coin, picking a card from a well-shuffled deck, etc. An example of a continuous uniform variable is the angle of the seconds-hand on a clock when you observe it at an arbitrary point in the day.

We use the notation $Y \sim \mathcal{U}(a, b)$ to designate the uniform distribution on $[a, b]$ to a random variable Y. Whether Y is continuous or discrete is usually left unspecified, and should be clear from context.

1.13.6 Gaussian (normal) distribution $\mathcal{N}(\mu, \sigma^2)$

The Gaussian or normal distribution is widely used in engineering and the sciences to model quantities whose value can in principle be any real number, but is expected to fall near a particular value. An fundamental instance of this are typical measurement errors. For example, a mill that fabricates 8-foot lengths of 2"x4" lumber can be expected to produce some pieces that are slightly longer and others that are slightly shorter than 8 feet. In this case, the length can be modeled as a Gaussian random variable with a mean of 8 feet.

The sample space for a univariate Gaussian distribution is the real numbers ($\Omega = \mathbb{R}$). The Gaussian family of distributions is parameterized with two numbers: μ and σ^2. μ is allowed to be any real number, while σ^2 is required to be positive (non-zero). Here is the formula for a Gaussian pdf:

$$p(y; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \tag{1.160}$$

This is shown in Figure 1.26 for several values of μ and σ^2. The notation for defining a normal variable is $Y \sim \mathcal{N}(\mu, \sigma^2)$. Next we prove that μ and σ^2 turn out to be the mean and variance of $\mathcal{N}(\mu, \sigma^2)$.

Theorem With $Y \sim \mathcal{N}(\mu, \sigma^2)$, $E[Y] = \mu$ and $Var[Y] = \sigma^2$.

Proof. Applying the definition from Eq. 1.26 to Eq. 1.160,

$$E[Y] = \int_{-\infty}^{\infty} y \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) dy \tag{1.161}$$

Change of variables: $z = \frac{y-\mu}{\sqrt{2\sigma^2}}$. Then $dz = \frac{1}{\sqrt{2\sigma^2}} dy$ and $y = \sqrt{2\sigma^2} z + \mu$.

$$E[Y] = \int_{-\infty}^{\infty} (\sqrt{2\sigma^2} z + \mu) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-z^2\right) \sqrt{2\sigma^2} dz \tag{1.162}$$

$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} (\sqrt{2\sigma^2} z + \mu) \exp\left(-z^2\right) dz \tag{1.163}$$

$$= \frac{1}{\sqrt{\pi}} \left(\sqrt{2\sigma^2} \int_{-\infty}^{\infty} z \exp\left(-z^2\right) dz + \mu \int_{-\infty}^{\infty} \exp\left(-z^2\right) dz\right) \tag{1.164}$$

Noting that $-1/2 \exp(-z^2)$ is the antiderivative of $z \exp(-z^2)$, we find that the first term in Eq. 1.164 equals $-1/2 \exp(-z^2)|_{-\infty}^{\infty} = 0$. For the second term, we use the result (without proving it), that $\int_{-\infty}^{\infty} \exp(-z^2) dz = \sqrt{\pi}$. Therefore,

$$E[X] = \frac{1}{\sqrt{\pi}} \left(0 + \mu\sqrt{\pi}\right) = \mu \tag{1.165}$$

For $Var[Y] = \sigma^2$ see https://proofwiki.org/wiki/Variance_of_Gaussian_Distribution. □

Here are two important properties of normal random variables (stated without proof).

- If X and Y are normal, then $Z = X + Y$ is normal.
- If X and Y are normal and uncorrelated, then they are independent.

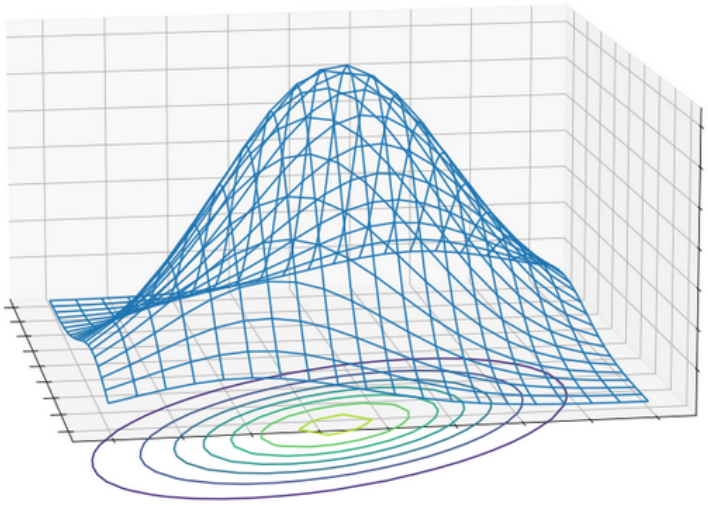

Figure 1.27: 2D multivariate normal.

Multivariate Gaussian variables

A multivariate Gaussian random variable $Y = (Y_1, \ldots, Y_n)$ is one whose univariate marginals are all Gaussian: $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ for $i = 1, \ldots, n$.

$$Y = (Y_1, \ldots, Y_n) \sim \mathcal{N}(\mu, \Sigma^2) \tag{1.166}$$

Here $\mu \in \mathbb{R}^n$ is the mean, and $\Sigma^2 \in \mathbb{R}^{n \times n}$ is the covariance matrix, which, like all covariance matrices, is symmetric and positive-definite. Figure 1.27 shows the bell-shaped pdf of a two-dimensional normal random variable. The level-sets (horizontal slices) of the pdf are concentric ellipses in the sample space (horizontal plane). These are centered on μ.

The formula for the multivariate pdf can be found in the Wikipedia article titled 'Multivariate normal distribution', but we will not be needing it. The important point is that it describes a set of measurements, each of which is normally distributed when observed in isolation. This does not mean that the measurements are pair-wise independent. In fact, two of the measurements could be perfectly correlated or even identical.

1.14 Central limit theorem

The central limit theorem (CLT) is a very important result in probability theory, as is establishes the Gaussian distribution as the common limit for all averaging processes. To understand the CLT, consider the length of my walking stride. This quantity is not Gaussian. It is usually positive since I rarely walk backwards. Also I generally have two walking modes: walking and trotting, so we expect the distribution of my stride to be "bi-modal", with two humps, as shown in Figure 1.28.

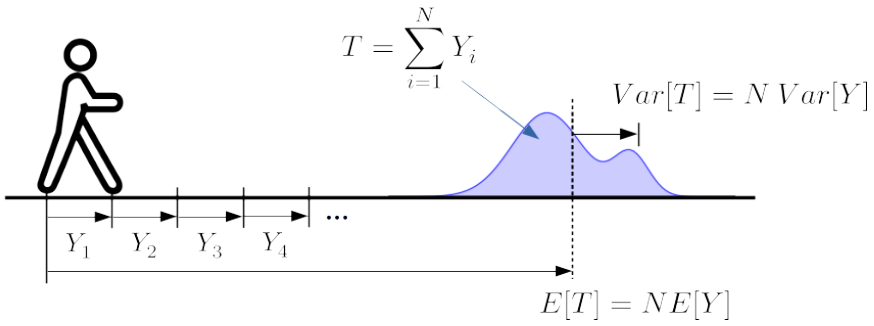

Figure 1.28: Central limit theorem.

Now imagine I take N independently sampled steps $\{Y_i\}_N \overset{\text{iid}}{\sim} Y$. The total distance I will advance is captured by the random variable T.

$$T = \sum_{i=1}^{N} Y_i \tag{1.167}$$

By the linearity of the expectation (Eq. 1.27),

$$E[T] = \sum_{i=1}^{N} E[Y_i] = NE[Y] \tag{1.168}$$

By the formula for the variance of a sum (Eq. 1.35), and using the iid assumption,

$$Var[T] = \sum_{i=1}^{N} Var[Y_i] = N\,Var[Y] \tag{1.169}$$

These results give us the mean and variance of T in terms of the mean and variance a single step, and the number of steps. However they don't give any information about the *shape* of the distribution of T. The central limit theorem states that, regardless of the shape of Y (caveats noted in the theorem), T will become "bell-shaped" (i.e. Gaussian) as N increases.

We must first define the *sample mean* ($\bar{Y}_N$) of an iid sample of Y ($\{Y_i\}_N \overset{\text{iid}}{\sim} Y$).

$$\bar{Y}_N = \frac{1}{N} \sum_{i}^{N} Y_i \tag{1.170}$$

Central limit theorem - Version 1.
Y is a random variable with finite variance. Then,

$$\bar{Y}_N \to \mathcal{N}\left(\mu_Y, \frac{\sigma_Y^2}{N}\right) \quad \text{as} \quad N \to \infty \tag{1.171}$$

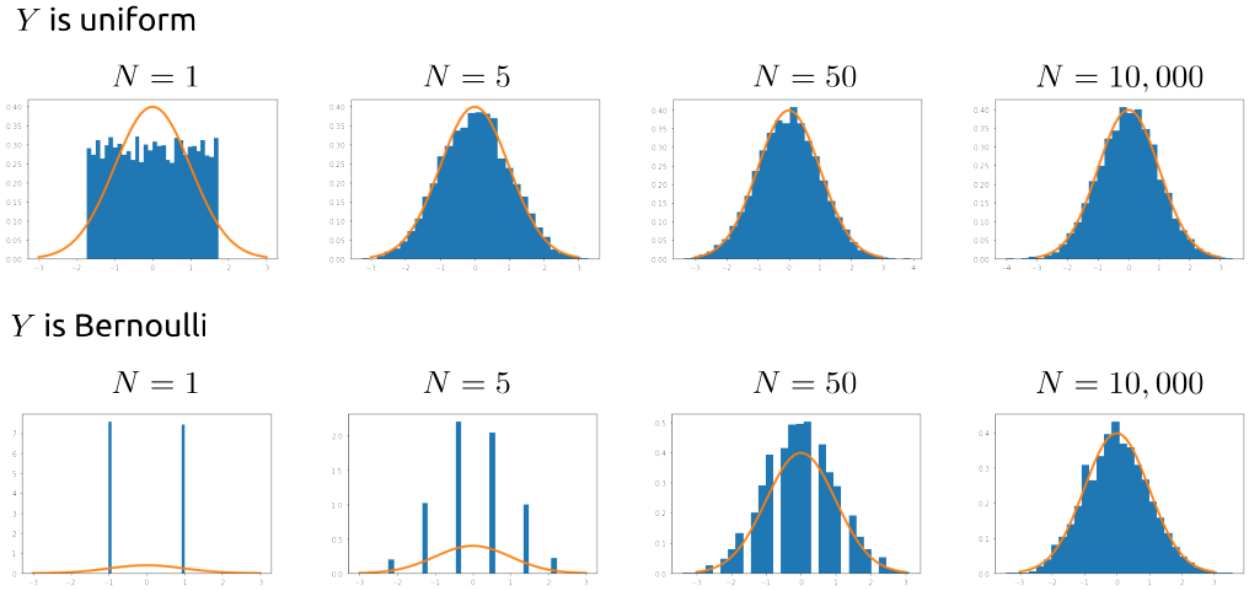

Figure 1.29: Simulation of the CLT.

In words, the sample mean becomes normal as N grows. The CLT asserts that the total distance I'll travel will become more normally distributed the farther I go. The effect of the bi-modality of my stride will eventually wash away. This is because every instance of a one-million-step experiment will have similar portions of walking and trotting modes.

It is also common to state the CLT in terms of a normalized version of the sample mean Z_N (this is known as the Z-statistic, it will resurface in Chapter 3).

$$Z_N = \frac{\bar{Y}_N - \mu_Y}{\sigma_Y/\sqrt{N}} \tag{1.172}$$

Central limit theorem - Version 2.
Y is a random variable with finite variance. Then,

$$Z_N \to \mathcal{N}(0,1) \quad \text{as} \quad N \to \infty \tag{1.173}$$

Figure 1.29 shows simulations that demonstrate the CLT. Each plot in the figure is a histogram of Z_N obtained with either a uniform (top row) or Bernoulli (bottom row) distribution for Y. The CLT predicts that, as N grows, these histograms will approach the standard normal distribution ($\mathcal{N}(0,1)$) shown in orange. Indeed, the uniform distribution reaches "normality" after only a handful of samples, whereas for the Bernoulli distribution it takes several thousand.

Chapter 2

Optimization theory

In this course we will learn several data-based techniques for building models of systems. Many of the techniques will follow a common paradigm: first we propose a parameterized family of models, then we choose the member of that family that best fits the data, according to some criterion. The search for the best-fitting model will be cast as an *optimization problem*, and we must therefore establish some of the basic concepts of optimization theory to describe and understand the techniques.

Optimization problems arise whenever we are faced with the task of choosing a *best* option from a set of possible options. This is an extremely broad formulation, and indeed optimization theory is useful in many different settings. Within engineering it can be applied to problems as wide ranging as these:

- What shape should we give a part such that its cost in minimized while meeting a specification?
- What voltage should we apply to each of the motors of a drone in order to stabilize its flight?
- How should the weights of a neural network be set?

2.1 Problem formulation

The specification of an optimization problem has three parts.

1. The *decision vector* x is an d-dimensional array of decision variables. Each of the x_i's can be real- or discrete-valued.
2. The *search set* or *feasible set* $\Omega \subseteq \mathbb{R}^d$ (not to be confused with the sample space from the previous chapter). This is the set of permissible values for the decision variables. Ω will be specified as a set of *equality* and *inequality constraints* on $\mathbb{R}^D$.
3. The *objective function* $J : \Omega \to \mathbb{R}$. This function assigns to each feasible decision vector $x \in \Omega$ a measure of "badness".

Our goal will be to find the "least bad" x in Ω, that is, the one that minimizes $J(x)$. The notation for the problem formulation is:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & J(x) \\ \text{subject to:} \quad & x \in \Omega \end{aligned} \tag{2.1}$$

Ω is specified as a set of n equality constraints $f_i(x)$ and m inequality constraints $g_j(x)$:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & J(x) \\ \text{subject to:} \quad & f_i(x) = 0 \quad i = 1 \dots n \\ & g_j(x) \leq 0 \quad j = 1 \dots m \end{aligned} \tag{2.2}$$

Using "argmin" in place of "minimize" returns the optimal decision vector x^*.

$$\begin{aligned} x^* \quad = \quad \underset{x}{\text{argmin}} \quad & J(x) \\ \text{subject to:} \quad & f_i(x) = 0 \quad i = 1 \dots n \\ & g_j(x) \leq 0 \quad = 1 \dots m \end{aligned} \tag{2.3}$$

We arbitrarily chose to cast the problem as a minimization problem, but could equally as well have described it as a maximization of a measure of "goodness".

Example 2.1.1. Suppose you wish build a rectangular enclosure for your pet rabbit using a fixed length ℓ of fence material, and you would like to know the width w and depth δ that maximize the area of the rectangle. This can be posed as an optimization problem with decision variable $x = (w, \delta)$. The objective is to maximize the function $J(w, \delta) = w\delta$. The feasible values are all positive w and δ that add up to a perimeter of ℓ. Here is the formulation of this problem as a mathematical optimization problem.

Given ℓ,

$$\begin{aligned} \underset{w,\delta}{\text{maximize}} \quad & w\delta \\ \text{subject to} \quad & 2w + 2\delta = \ell \\ & w \geq 0 \\ & \delta \geq 0 \end{aligned} \tag{2.4}$$

We can convert the problem to a minimization simply by flipping the sign of the objective function. Objective functions in minimization problems are usually called "cost" functions.

$$\begin{aligned} \underset{w,\delta}{\text{minimize}} \quad & -w\delta \\ \text{subject to} \quad & 2w + 2\delta = \ell \\ & w \geq 0 \\ & \delta \geq 0 \end{aligned} \tag{2.5}$$

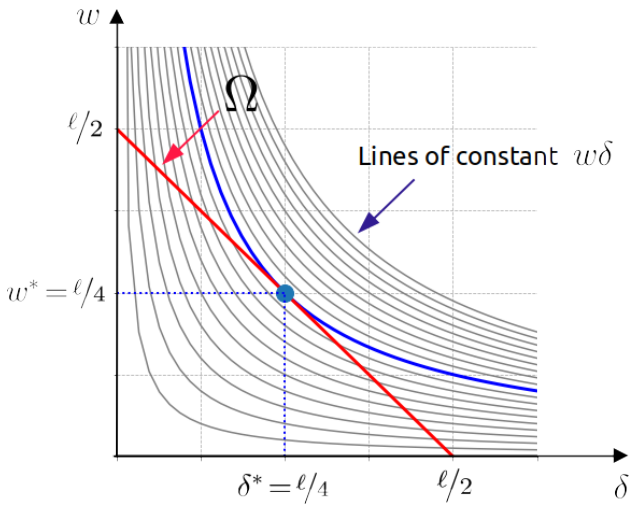

Figure 2.1: The feasible set is the red line segment, which is the restriction of the line $2w + 2\delta = \ell$ to the positive quadrant. The solution (w^*, δ^*) must lie on this line. The cost function $J = -w\delta$ is a surface that dips into the page, and descends from the bottom left corner to the upper right corner. Its level sets are the gray curves. The lowest point along this surface on the red line is at the blue dot, i.e. at $w^* = \delta^* = \ell/4$. Thus, the optimal shape is a square.

This problem has $d = 2$ decision variables, $n = 1$ equality constraint, and $m = 2$ inequality constraints. Figure 2.1 provides an illustration.

2.1.1 Global vs. local solutions

The optimal decision vector x^* is known as the *global solution* to the problem. In the previous example there was only one global solution, but there could be many. A global solution is any $x^* \in \Omega$ that satisfies

$$J(x^*) \leq J(x) \qquad \forall\, x \in \Omega \tag{2.6}$$

It is also possible that a problem has *no* global solution. For example the minimization of $J(x) = x$ over the real numbers has no solution, since there is no smallest real number.

A weaker sense of solving an optimization problem is to find a *local solution*. This is a feasible point (a.k.a. decision vector) that is best amongst its immediate feasible neighbors, but not necessarily best overall. For many problems, this will be the best we can do. A vector x^+ is a local solution if,

$$J(x^+) \leq J(x) \qquad \forall\, x \in \Omega \cap B_\epsilon(x^+) \tag{2.7}$$

where $B_\epsilon(x^+)$ is some neighborhood of x^+.

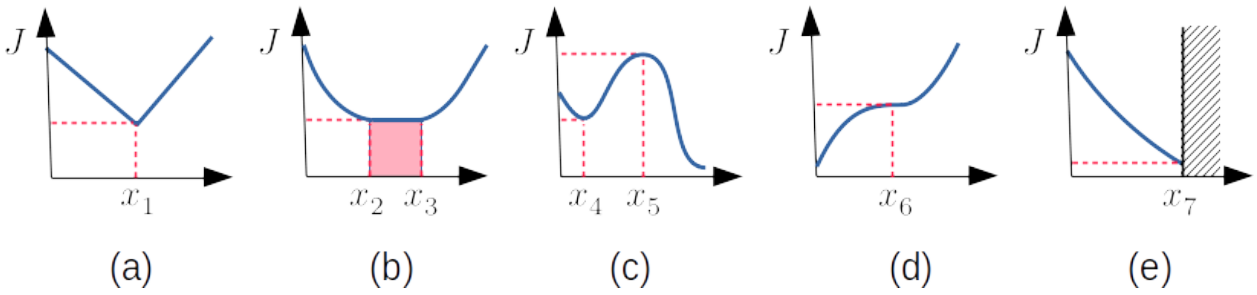

Figure 2.2: Categorizing feasible points

2.2 Types of feasible points

Next we list three ways of categorizing feasible points: interior vs. non-interior, differentiable vs. non-differentiable, and stationary vs. non-stationary.

1. **Interior vs. non-interior points.** As the name suggests, an interior point of a set is one that is located "inside" the set. All other points are non-interior. The formal definition of an interior point is one with a neighborhood that is entirely contained in the set. We will use Ω° for the interior of Ω (the collection of all interior points).

   We can characterize Ω° in terms of the constraints of the problem. Ω° is the set of points for which no constraint is *active*. An active constraint is one in which the relation is satisfied with the "=" symbol. Equality constraints are always active. An inequality constraint $g_j(x) \leq 0$ is active at x if $g_j(x) = 0$. Any feasible set with equality constraints ($n \geq 1$) has an empty interior.

2. **Differentiable vs. non-differentiable points.**

   We say that a point x is a *differentiable point* when the cost function J is continuously differentiable at x. This means that the gradient of J, denoted with ∇J exists and is continuous at x. Otherwise x is a *non-differentiable point*. The gradient is a generalization of the scalar derivative to functions with multiple inputs. ∇J is a vector in $\mathbb{R}^d$ that points in the direction of most rapid increase of J with respect to changes in x.

3. **Stationary vs. non-stationary points.** A point $x \in \Omega$ is a *stationary point* of J when $\nabla J(x) = 0$. This means that the function does not increase nor decrease in any direction. It is locally flat. Stationary points are important points to consider when solving optimization problems, as we will see in the next section.

Example 2.2.1. The feasible set in the enclosure example has no interior, because it has an equality constraint.

Figure 2.2 illustrates these concepts. Plot (a) shows a non-differentiable point x_1. This is *not* a stationary point, since the gradient is not defined at x_1. In (b) there is a continuum of stationary points between x_2 and x_3. All are interior points, differentiable, and also global

minima. Plot (c) shows two stationary points x_4 and x_5. x_4 is a local solution but not a global solution, x_5 is not a local solution. Plot (d) shows another example of a stationary point that is neither a local nor a global solution. Finally, in plot (e), x_7 is a non-stationary non-interior point that is both a local and a global solution.

From these plots we can begin to see that the solutions to optimization problems are of at least three types:

1. non-differentiable points, as in (a),

2. non-interior points, as in (e), and

3. stationary points, as in (b).

The *first order condition for optimality* establishes that these are in fact the *only* possibilities.

2.2.1 First order optimality condition

The statement of the first order condition for optimality is as follows.

$$x \text{ is a differentiable, interior, local solution} \quad \Rightarrow \quad x \text{ is stationary} \tag{2.8}$$

At first glance, the statement may not seem very useful. It says that, if we know that a point is a local solution, as well as differentiable and interior, then we can assert that it is stationary. However it is much easier to test for stationarity than for local optimality; just evaluate the gradient. A better way to interpret this is to notice that it implies that all local solutions are to be found amongst three types of points: 1) stationary points, 2) non-differentiable points, and 3) non-interior points. This is because all *points* are either 1) differentiable and interior, or 2) non-differentiable, or 3) non-interior (some may be both 2 and 3). And if a point in category 1) is a solution, then the first order condition tells us it must be stationary.

The first order condition has its largest impact when J is continuously differentiable everywhere and the problem has no constraints. Then, all points are differentiable and interior, and the condition reduces to:

$$x \text{ is a local solution} \quad \Rightarrow \quad x \text{ is stationary} \tag{2.9}$$

Add to this the fact that all global solutions are local solutions.

$$x \text{ is a global solution} \quad \Rightarrow \quad x \text{ is a local solution} \quad \Rightarrow \quad x \text{ is stationary} \tag{2.10}$$

In this context, stationarity is a *necessary but not sufficient* condition for local and hence global optimality. Non-sufficiency is demonstrated by points x_5 and x_6 of Figure 2.2, which are both stationary but not local solutions. Despite not being sufficient, the condition suggests a procedure for solving differentiable/unconstrained optimization problems. First, find all of the stationary points (by solving $\nabla J(x) = 0$). Then, assuming there are a finite

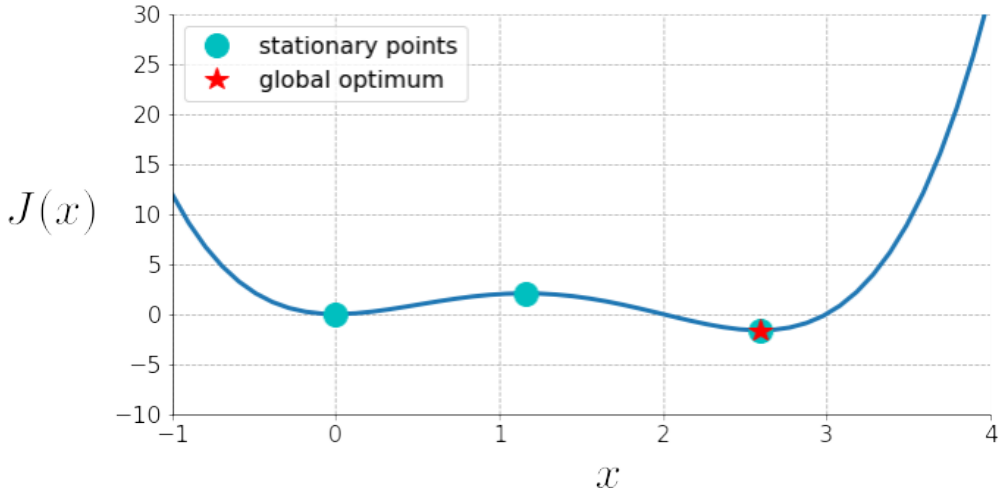

Figure 2.3: Example 2.2.2.

number of such solutions, evaluate J for each of them, and choose the minimizer. The following example demonstrates this procedure in 1D.

Example 2.2.2. Find the minimum of the function $J(x) = x^3(x-3)(x-2)$.
Solution. A plot of $J(x)$ is shown in Figure 2.3. We begin by computing the derivative of $J(x)$.

$$\nabla J(x) \;=\; \frac{dJ}{dx}(x) \;=\; \frac{d}{dx}\left(x^2(x-3)(x-2)\right) \;=\; 4x^3 - 15x^2 + 12x \tag{2.11}$$

$\nabla J(x)$ is continuous everywhere, so $J(x)$ is continuously differentiable everywhere. Since the problem is also unconstrained, we are assured by the first order necessary condition that any local solution must be stationary. The roots of $\nabla J(x)$ can be found with the standard formula for quadratic equations, or using Python, and they are $\{0, 1.16, 2.60\}$ (green dots in the figure). Finally we evaluate J on each of the stationary points and choose the one with the least value: $x^* = 2.60$ (red star in the figure).

When presented with an optimization problem, there are some important questions to consider:

1. What is its size? That is, how many decision variables and constraints does it have? Both the dimension and the number of constraints strongly influence the amount of computation and time needed to solve a problem.

2. Are the decision variables real or integer-valued? Integer-valued problems are harder to solve than real-valued problems, since many numerical methods rely on the gradient. Fortunately, the problems that we will encounter in this course all involve real-valued decision variables and differentiable objective functions.

3. Is the problem convex? The first order conditions become "supercharged" if we can establish that the problem is *convex*.

2.3 Convex optimization problems

Convex optimization problems are ones with a special structure that makes them relatively easy to solve. All other problems are non-convex. Non-convex problems are usually difficult to solve in the global sense, although they can sometimes be solved in the local sense using the first order condition.

A *convex optimization problem* is a minimization problem in which both the feasible set Ω and the cost function J are *convex*. The definitions of a *convex set* and a *convex function* are given next.

- A set is convex if for any two of its elements a and b, the line segment $\overline{ab}$ is entirely contained in the set.
- A function is convex if its epigraph is a convex set. The epigraph of a function is the set of points that lay above the graph of the function. The epigragh of $f(x)$ is $\text{epi}(f) = \{(x, y) \mid y \geq f(x)\}$.

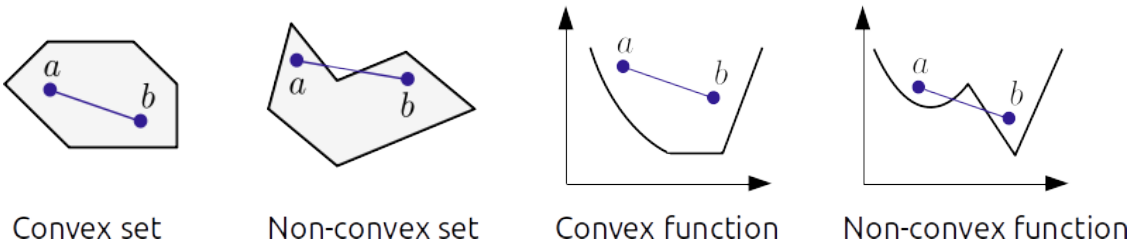

Figure 2.4: Convex sets and functions

These concepts are illustrated in Figure 2.4. On the left we see convex and non-convex sets. For the convex set, no line segment that begins and ends within the set, leaves the set. The non-convex set has a "dimple", which violates convexity. Convex functions are bowl-shaped. Their epigraph (the region above the function) is a convex set. A convex optimization problem is therefore one with a dimple-less feasible set and a bowl-shaped cost function.

2.3.1 Properties of convex optimization problems

There are two important facts about convex problems that make them easier to solve.

1. For convex problems, every local solution is a global solution. The practical implication of this is that we can use local solvers (e.g. gradient descent) to find global solutions.
2. For convex problems with continuously differentiable cost, stationary points are local solutions. That is, situations such as that of points x_5 and x_6 in Figure 2.2 do not arise.

Together, these add left-facing arrows to Eq. 2.10, which now becomes:

$$x \text{ is a global solution} \iff x \text{ is a local solution} \iff x \text{ is stationary}$$

Hence, for unconstrained smooth convex problems, the set of stationary points is the same as the set of global solutions. To solve such problems we need only to solve the system of equations $\nabla J(x) = 0$. We may, for simple problems, be able to find a solution analytically. In practice we use numerical methods such as Newton's method, or gradient descent. Newton's method can be faster, but it requires knowledge of the Hessian of J (i.e. its second derivative). The gradient descent method requires knowledge only of ∇J, and it is the method that we will use in this course (Section 2.4).

Next we list a few important instances of convex sets and convex functions.

2.3.2 Examples of convex sets

Euclidean norm ball

The Euclidean norm of a vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, also known as the 2-norm, is a generalization of the standard 3D notion of distance to an d-dimensional vector space. We denote it with $\|x\|_2$.

$$\|x\|_2 = \left(\sum_{i=1}^{d} x_i^2 \right)^{1/2} \tag{2.12}$$

Notice that in three dimensions, this is the straight-line distance from the origin to x: $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2}$. A Euclidean norm ball (a.k.a. the 2-norm ball) is a generalization of a 3D sphere. It is defined as $\{x \in \mathbb{R}^d : \|x\|_2 \leq r\}$, where r is the radius of the ball. The Euclidean norm ball is convex.

p-norm balls

The Euclidean norm can itself be generalized by replacing the 2's in Eq. 2.12 with p's, where p is a positive integer. With $p \geq 1$, the p-norm is defined as,

$$\|x\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p} \tag{2.13}$$

The p-norm ball is analogous to the Euclidean norm ball,

$$\{x \in \mathbb{R}^d : \|x\|_p \leq r\} \tag{2.14}$$

and it is also convex. Of course, nothing prevents us from using $p < 1$, however the resulting function fails to be a norm, and its corresponding ball fails to be convex.
Figure 2.5 shows some examples of p-norm balls in $\mathbb{R}^2$. Note three important cases.

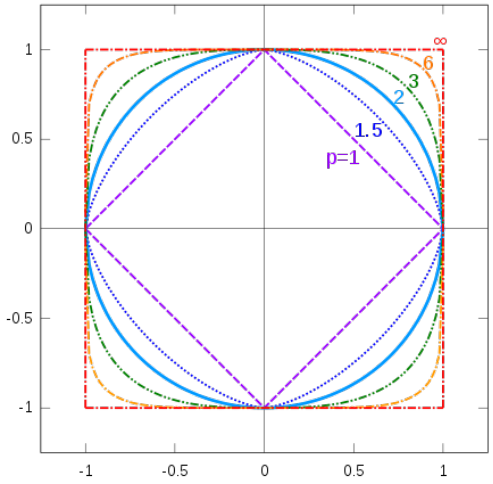

Figure 2.5: p-norm balls with unit radius (Image from Wikipedia)

- **1-norm.** With $p = 1$ the ball appears diamond-shaped. This is the Manhattan, or taxicab norm, so-called because it measures distance only along vertical and horizontal displacements (like a taxicab driving through Manhattan).

- **2-norm.** The Euclidean norm ball is a circle.

- **∞-norm.** As p goes to infinity, the p-norm ball approaches a square, and Eq. 2.13 returns the largest absolute value among the components of x.

$$\|x\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|) \tag{2.15}$$

Affine equality constraints

An affine equality constraint is a formula of the form,

$$\alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_d x_d = \beta \tag{2.16}$$

Here the α_i's and β are real numbers, and the x_i's are the decision variables. The set of points that satisfy this formula is called a *hyperplane*, and it is the generalization of a 3D plane to d dimensions. We can arrange n such affine equality constraints into a matrix form.

$$Ax = b \tag{2.17}$$

where A is an $n \times d$ matrix whose coefficients are the α's, and b is a $n \times 1$ column vector with the β's. The set of points that satisfy Eq. 2.17, or equivalently, the intersection of n hyperplanes, is convex.

Convex inequality constraint

A convex inequality constraint is a formula of the form,

$$g(x) \leq 0 \tag{2.18}$$

where $g(x)$ is a convex function. The set of points x that satisfy a convex inequality constraint is convex. Because the intersection of any number of convex sets is also convex, we find that the specification of any number of convex inequality constraints defines a convex feasible set. This leads us to an important fact.

The constraint set of a convex optimization problem consists of affine equalities and convex inequality constraints

2.3.3 Convex functions

Here are some examples of convex functions. In each case, J is a function from $\mathbb{R}^d$ to $\mathbb{R}$.

- **Affine functions**. $J(x) = a \cdot x + b$, with $a \in \mathbb{R}^d$, $b \in \mathbb{R}$.
- **p-norms**. $J(x) = \|x\|_p$ with $p \geq 1$.
- **Function composition**: $J(x) = g(h(x))$ is convex whenever h convex and g is affine. Here h is a function that takes $x \in \mathbb{R}^d$ and returns a real number, and g takes that real number and returns another real number. In other words, convexity is preserved by composition with an affine scalar function.

These examples will reappear as cost functions later in the course.

2.4 Gradient descent

Gradient descent is a numerical technique for minimizing differentiable, real-valued functions $J : \mathbb{R}^d \to \mathbb{R}$. The method can be understood by imagining the function as a hilly terrain, as depicted in Figure 2.6. The algorithm advances like a person walking along the terrain in search of its lowest point. The person can only look down at the ground – they cannot raise their sight to look around. At each moment, they observe the slope of the ground under their feet, and move in the downward direction. They continue in this way until they reach an area that is flat, i.e. a local minimum.

Gradient descent advances in the direction of the *negative gradient*. In our two-dimensional example, the gradient is the blue arrow in the horizontal plane, pointing in the direction of steepest ascent. Its negative (the green arrow) indicates the direction of steepest *descent*.

The gradient descent algorithm begins with the arbitrary selection of a starting point $x_0 \in \Omega$. From there, it proceeds by taking steps to x_1, x_2, ... until no more downward progress can be made. Then it returns its last value x_K. If the problem is unconstrained,

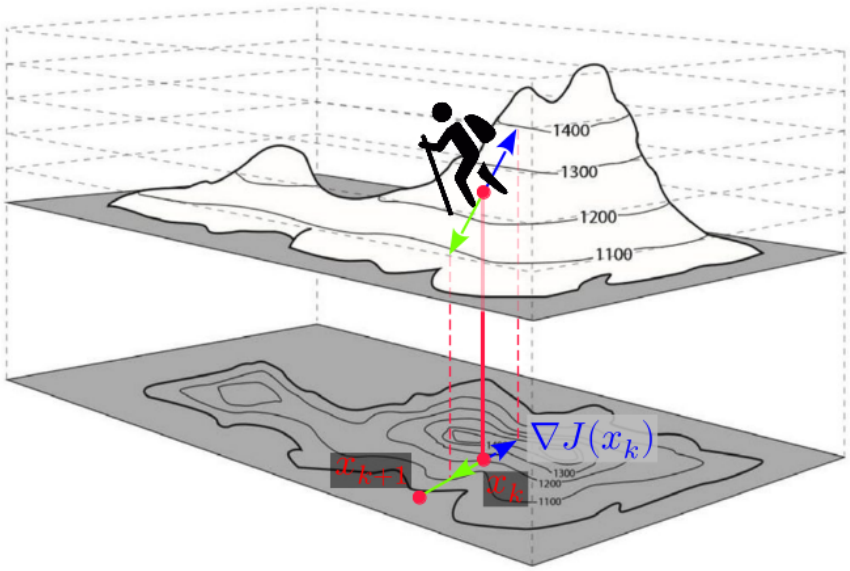

Figure 2.6: The gradient descent method

then we can conclude that the algorithm has reached a stationary point. If furthermore the problem is convex, then we have found a global optimum.

The update rule for gradient descent is,

$$x_{k+1} = x_k - \gamma \nabla J(x_k) \tag{2.19}$$

Here x_k is a candidate solution after k steps and γ is the *step size* parameter. The step size can be kept fixed, or it can be varied with each step, either in a predetermined manner or in a way that depends on the current value of the gradient. There are strategies for varying γ that guarantee convergence to a local minimum. There are also bad choices for γ that prevent convergence. Gradient descent does not guarantee convergence to a local minimum unless the step size parameter is properly chosen.

Although we will not study them here, it is worth knowing that there are extensions of gradient descent for problems with constraints. Equality constraints are typically treated by appending them to the cost function using *Lagrange multipliers*. For inequality constraints, projected gradient methods prevent the solution from leaving Ω by projecting the gradient onto the boundary. Alternatively, the Frank-Wolfe algorithm finds feasible directions by solving intermediate convex programs.

Next we will introduce a variant of gradient descent that is useful for optimization problems that are typical of data-based modeling techniques: *stochastic gradient descent*.

2.4.1 Stochastic gradient descent (SGD)

As we will learn later in the course, many techniques in machine learning require us to find a solution to the following optimization problem.

Given $\mathcal{D} = \{y_i\}_N \overset{\text{iid}}{\sim} Y$, find

$$\underline{\theta}^* = \underset{\underline{\theta}}{\operatorname{argmin}}\ E[L(Y\ ;\ \underline{\theta})] \tag{2.20}$$

The dataset $\mathcal{D}$ is an iid sample of a (possibly multivariate) random variable Y. The decision variable $\underline{\theta} \in \mathbb{R}^d$ is an array of parameters corresponding to the chosen parametric family. The goal is to find parameter values that minimizes the expected value of the *loss function* L. This will be defined in Chapter 4. The loss is itself a random variable, since it is a function of Y. The is an example of a *stochastic optimization problem*; one in which the cost function is uncertain.

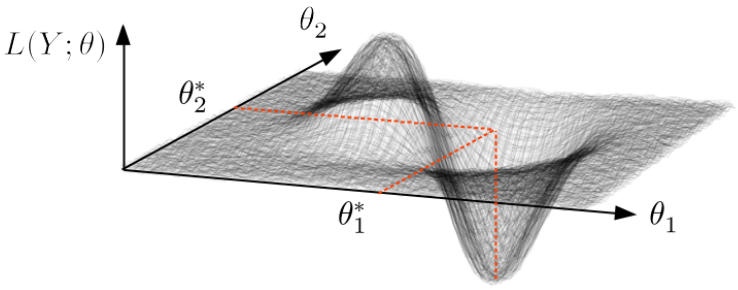

Figure 2.7: Stochastic cost function

Figure 2.7 provides an illustration. The search space Ω is the set of feasible parameters, which we assume to be unconstrained. For each value of $\underline{\theta} = (\theta_1, \theta_2)$, the variations in Y produce variations in $L(Y;\underline{\theta})$. If our goal is to build a model that will work well in the long run, then the law of large numbers suggests that we should minimize the expected value of the loss. The main obstacle to doing this is that, because we do not know the distribution of Y, we cannot know the distribution of L.

To get around this problem, we approximate the expectation using the sample mean.

$$E[L(Y\ ;\ \underline{\theta})] \approx \frac{1}{N}\sum_{i=1}^{N} L(y_i\ ;\ \underline{\theta}) \tag{2.21}$$

This leads to an approximation of the original problem.

$$\underline{\theta}^* = \underset{\underline{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i\ ;\ \underline{\theta}) \tag{2.22}$$

The multiplicative factor $1/N$ has been dropped because it does not affect the result. Applying

gradient descent (Eq. 2.19) to this problem produces the following update rule.

$$\underline{\theta}_{k+1} = \underline{\theta}_k - \gamma \nabla_{\underline{\theta}} \left(\sum_{i=1}^{N} L(y_i \,;\, \underline{\theta}) \right) \tag{2.23}$$

$$= \underline{\theta}_k - \gamma \sum_{i=1}^{N} \nabla_{\underline{\theta}} \, L(y_i \,;\, \underline{\theta}) \tag{2.24}$$

The notation $\nabla_{\underline{\theta}}$ indicates that the gradient is taken with respect to $\underline{\theta}$. Eq. 2.24 is the direct application of plain-vanilla gradient descent to the stochastic optimization problem. This works well with small to medium-sized datasets, however for large datasets (large N), the performance is poor. This is because each step involves N evaluations of $\nabla_{\underline{\theta}} J$; one for each data point y_i. For large N, this means that a large amount of computation is needed for each step of gradient descent.

The idea behind SGD is simple: instead of basing the estimate of $\nabla_{\underline{\theta}} J$ on the full dataset $\mathcal{D}$, we use a reduced dataset $\mathcal{B}$, which we call a *batch*.

$$\underline{\theta}_{k+1} = \underline{\theta}_k - \gamma \sum_{y_i \in \mathcal{B}} \nabla_{\underline{\theta}} \, L(y_i \,;\, \underline{\theta}) \tag{2.25}$$

The batch size is a tuning parameter of the algorithm. However, since the sample mean is unbiased for all N, choosing a smaller batch will not affect the bias of the estimate, although it will increase its variance. The hope is that this increased variance is compensated by the more frequent steps taken by the algorithm, which allows it to advance quickly in the early stages.

Another parameter of SGD is the method by which we sample $\mathcal{B}$ from $\mathcal{D}$. Certainly we do not want any points in $\mathcal{D}$ to be ignored. Hence there are two reasonable approaches: sampling *with replacement* and *without replacement*. To sample with replacement means to choose each element of $\mathcal{B}$ randomly from the full $\mathcal{D}$. Sampling without replacement means that we partition $\mathcal{D}$ into an integer number K of batches ($K = {}^{|\mathcal{D}|}/_{|\mathcal{B}|}$), and we use one batch per step. A single K-step pass through $\mathcal{D}$ is called an *epoch*. Typically, SGD will run for many epochs.

The stopping criteria for SGD is a bit tricky. The noisy nature of the algorithm means that it will bounce around without ever settling down, and hence we cannot simply look at $\left\| \underline{\theta}_{k+1} - \underline{\theta}_k \right\|$ to decide when to stop. Rather, as SGD advances, we track the performance of the resulting model and stop when the performance begins to deteriorate, due to a phenomenon known as *overfitting*. We will delve deeper into these topics in lab and later in the course.

2.5 Gradient-less optimization

Gradient descent can only be used if $\nabla_{\underline{\theta}} J$ exists. We will encounter optimization problems of this type when we learn about *hyper-parameters* in Chapter 4. Here are a couple examples of algorithms that can be used to solve optimization problems in lieu of a gradient.

2.5.1 Grid search/Exhaustive search

This is a "brute force" method that evaluates the objective function on a large number of predetermined points in the feasible space, and returns the best one. The algorithm is called "grid search" when the decision variables are continuous, and "exhaustive search" when they are discrete.

Grid/exhaustive search can be inefficient since it makes no assumptions about the decision variables or the cost function. However it has the advantage that it is easy to understand and to implement in code. A simple implementation of grid/exhaustive search consists of of nested for-loops. The algorithm is also easily parallelized by sending each node of the grid to a separate unit of computation, and then gathering and finding the minimum (or maximum) amongst the results.

2.5.2 Genetic algorithms

Genetic algorithms can only be used when the decision variables are numerical (integer-valued or real-valued, but not labels). Under this assumption, they improve upon grid/exhaustive search by making smart decisions regarding which points to evaluate next, instead of naively evaluating all points. The approach is based on an analogy to Darwin's theory of evolution by natural selection. In the analogy, each point $x \in \Omega$ is an "individual". The objective function $J(x)$ returns the "fitness" of the individual (the terminology assumes a maximization problem). The goal is to "evolve" a population (a set of points in Ω), by raising its average fitness with each generation. Doing this for many generations can produce high fitness individuals, which are good solutions to the optimization problem. The details of genetic algorithms are beyond our scope, however you may choose to delve deeper into this topic with your class project.

Chapter 3

Statistical inference: Static models with no inputs

In this chapter we begin to apply the tools of probability and optimization theory from Chapters 1 and 2 to the construction of closed-box models. We'll start with the simplest case: models with no inputs. We will see that, despite the simplicity of the setup, the techniques that we will cover have wide application. Furthermore, some of the concepts introduced in this chapter will continue to play a role in the more general contexts of later chapters. Figure 3.1 illustrates the approach. We imagine the system to be a box that produces numbers according to a random variable Y, and we collect a dataset $\mathcal{D}$ consisting of N iid samples of Y.

$$\mathcal{D} = \{y_i\}_N \sim Y \tag{3.1}$$

Here $\{y_i\}_N$ denotes a set of N numbers indexed with i. The iid assumption is crucial since it guarantees that all of the samples come from the same distribution, and it precludes any complexities in the sampling process. We will drop this assumption when we study dynamical models in Chapter **??**, and replace it with assumptions that constrain the manner in which the samples can be correlated over time.

We will cover three important techniques from statistics: *point estimates*, *confidence intervals*, and *hypothesis tests*. Each of these address different types of questions, as described below.

1. **Point estimation** techniques produce a *best estimate* of a property or parameter of p_Y. The two generic *properties* of random variables that are most commonly estimated are the mean and the variance. For example, we may be interested in estimating the average value and variability in the compressive strength of a particular cement

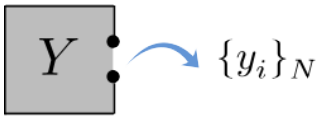

Figure 3.1: Sampling a dataset from an input-less system modeled as a random variable Y.

mixture. The mean and the variance are generic "properties" of all random variables, and we will see that there are generic formulas for estimating them that do not rely on assumptions about the *family* of p_Y. In contrast, one might be interested in estimating a *parameter* that is specific to a particular family of distributions. For example the maximum value b of a uniform distribution $\mathcal{U}(a, b)$. For this problem we will learn the technique of *maximum likelihood estimation*.

2. **Confidence interval problems**. Instead of a best value of a property or parameter, a confidence interval is a *range* of values that is believed to contain the property or parameter with a given *level of confidence* γ.

3. **Hypothesis tests** are used to decide between two theories about Y: a *null hypothesis* and an *alternate hypothesis*.

3.1 Point estimation

Point estimation seeks to compute a single best estimate of a parameter (or property) of a distribution, based on a dataset $\mathcal{D} = \{y_i\}_N$. We will denote the parameter with θ and assume for now that it is a scalar quantity (not a vector). At an abstract level, a point estimator is simply a function g_N that maps from the space of possible datasets to the space of possible parameter values.

$$g_N : \mathbb{R}^N \to \mathbb{R} \tag{3.2}$$

The value returned by the estimator when evaluated on a particular dataset is called an *estimate*, and it is denoted as $\hat{\theta}_N$.

$$\hat{\theta}_N = g_N(y_1, \ldots, y_N) \tag{3.3}$$

The first question that arises is how to decide whether a given estimator is "good". We will address this question in sections 3.1.1 through 3.1.5. Then, in section 3.1.6 we will see how the problem of generating an estimator can be addresed using optimization theory.

The mathematical approach to reasoning about the quality of an estimator begins with the definition of the iid set of random variables $\{Y_i\}_N \overset{\text{iid}}{\sim} Y$ (see Chapter 1.7). Passing these variables through the estimator function g_N produces a new random variable $\hat{\Theta}_N$.

$$\hat{\Theta}_N = g_N(Y_1, \ldots, Y_N) \tag{3.4}$$

$\hat{\Theta}_N$ is also (somewhat confusingly) called the "estimator". The distribution of $\hat{\Theta}_N$ describes the possible values of $\hat{\theta}_N$ due to variations in the possible datasets $\mathcal{D}$. Both g_N and $\hat{\Theta}_N$ perform the same function of estimating θ, but whereas g_N captures the computational properties of the estimator, $\hat{\Theta}_N$ captures its statistical properties. Our notion of the "quality" of the estimator will depend only on statistical properties, and hence we will focus on $\hat{\Theta}_N$, and not g_N. Next we define the two properties of $\hat{\Theta}_N$ that determine its quality as an estimator of θ. These are its *bias* and its *variance*.

3.1.1 Bias and Variance of an estimator

Imagine that you are given a function g_N that estimates the size of the largest apple in an apple orchard from a sample of $N = 100$ apples. Perhaps this function returns the largest number in the dataset,

$$g_N(y_1, \ldots, y_{100}) = \max_i y_i \tag{3.5}$$

or maybe it returns the maximum plus 10% of the difference between the largest and the smallest apple in the basket:

$$g_N(y_1, \ldots, y_{100}) = \max_i y_i + 0.1(\max_i y_i - \min_i y_i) \tag{3.6}$$

The possibilities are endless. Whichever estimation function you choose, you can use it repeatedly to estimate the largest apple size using different baskets of 100 apples. Each time, the resulting estimate will be slightly different, due to the natural variation in the sizes of apples in each basket.

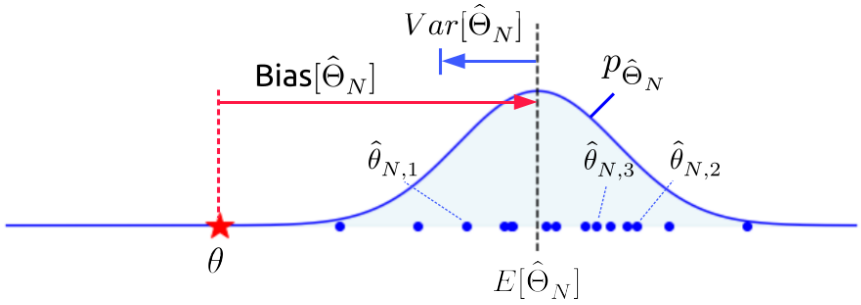

Figure 3.2: Bias and variance of an estimator

This is illustrated in Figure 3.2. The blue dots are estimates of maximum apple size made with g_N and with different baskets of 100 apples: $\mathcal{D}_1$ produces $\hat{\theta}_{N,1}$, $\mathcal{D}_2$ produces $\hat{\theta}_{N,2}$, and so on. Each of the $\hat{\theta}_{N,i}$'s can be regarded as a sample from $\hat{\Theta}_N$, whose pdf is shown in blue. The shape of the distribution $p_{\hat{\Theta}_N}$ depends both on the distribution of Y (the orchard), and on the estimator function g_N.

The true value of the parameter θ is indicated in the figure with a star. In this case, the estimator severely overestimates the size of the largest apple. This shows up in the figure as as a large positive estimator *bias*. Mathematically, the bias of an estimator is defined as the difference between its expected value and the true value of the parameter:

$$\text{Bias}[\hat{\Theta}_N] = E[\hat{\Theta}_N] - \theta \tag{3.7}$$

The bias measures the average error of the estimator. A positive bias means that the estimator overestimates θ; a negative bias means that it underestimates θ. All other things being equal, we prefer an estimator with a smaller absolute value of bias. In general, the bias of an estimator depends on g_N, on the distribution of Y, and on the value of θ. Since θ is unknown, it is not generally possible to compute the numerical value of the bias. However

we can often express the bias *as a function of* θ, and in the best case it may be possible to show that the bias equals zero for all values of θ. If so, we say that the estimator is *unbiased*.

The *variance* of an estimator is simply the variance of $\hat{\Theta}_N$, as shown in Figure 3.2.

$$\text{Var}[\hat{\Theta}_N] = E\left[\left(\hat{\Theta}_N - E[\hat{\Theta}_N]\right)^2\right] \tag{3.8}$$

The variance quantifies the sensitivity of the estimator to natural variations in the data. An estimator with high variance may, for example, return wildly different estimates of maximum apple size for each basket of apples. All other things being equal, we prefer an estimator with a *smaller* variance – i.e. one that is more robust to variations in the data. Unlike the bias, the variance does not depend on the unknown value of θ, and can therefore be estimated from the data.

Later we will describe higher level criteria that build upon the bias and variance. First we introduce some simple but useful examples: the *sample mean* and the *sample variance*.

3.1.2 Point estimation of the mean

Recall the definition of the mean of a random variable.

$$E[Y] = \int_{\Omega_Y} y\, p_Y(y) dy \tag{3.9}$$

And recall that $E[Y]$ is also denoted as μ_Y. Consider the problem of estimating μ_Y from a dataset $\mathcal{D} = \{y_i\}_N \overset{\text{iid}}{\sim} Y$. A straightforward approach is simply to take the average of the data. This is called the *sample mean estimate*:

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{3.10}$$

We use the symbol $\bar{Y}_N$ for its corresponding estimator:

$$\bar{Y}_N = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{3.11}$$

Taking the average of the data certainly *seems* like a reasonable way of estimating μ_Y. Let's verify that this is so by computing the bias and variance of $\bar{Y}_N$.

Bias of the sample mean

The sample mean is an *unbiased* estimator, meaning that its bias equals zero for all possible distributions Y (provided $E[Y]$ is finite). The proof of this is simple.

$$\begin{aligned}
\text{Bias}[\bar{Y}_N] &= E[\bar{Y}_N] - \mu_Y && (3.12)\\
&= E\left[\frac{1}{N}\sum_{i=1}^{N} Y_i\right] - \mu_Y && (3.13)\\
&= \frac{1}{N}\sum_{i=1}^{N} E[Y_i] - \mu_Y && (3.14)\\
&= \left(\frac{1}{N}\sum_{i=1}^{N} \mu_Y\right) - \mu_Y && (3.15)\\
&= 0 && (3.16)
\end{aligned}$$

The first two lines are obtained by definition. The third line is obtained by the linearity of the expected value. The fourth is by the iid assumption. Notice that the result does not depend on N. Even the extreme case of $N = 1$ yields an unbiased estimator. The benefit of a larger sample size is reflected in the variance of $\bar{Y}_N$.

Variance of the sample mean

We can compute the variance of the sample mean in terms of the variance of Y.

$$\begin{aligned}
Var[\bar{Y}_N] &= Var\left[\frac{1}{N}\sum_{i=1}^{N} Y_i\right] && (3.17)\\
&= \frac{1}{N^2}\sum_{i=1}^{N} Var[Y_i] && (3.18)\\
&= \frac{\sigma_Y^2}{N} && (3.19)
\end{aligned}$$

The second line is obtained by the rule for the variance of a linear combination (Eq. 1.35). The third line is by the iid assumption. The variance of the sample mean is inversely proportional to size of the sample. The larger the sample, the lower the variance of the estimator, the more certain we are that the true mean is near the average of the data.

3.1.3 Point estimation of the variance

Next we develop an estimator for σ_Y^2, the variance of Y. Recall the definition of the variance as the mean of the squared deviation from the expected value.

$$\sigma_Y^2 = E[(Y - \mu_Y)^2] \tag{3.20}$$

We can construct an estimator of the variance by replacing μ_Y in this equation with the sample mean, and approximating the expectation of $(Y - \hat{\mu}_N)^2$:

$$\hat{\sigma}_N^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \hat{\mu}_N)^2 \tag{3.21}$$

Notice the $N-1$ in the denominator, where the definition of the sample mean would have an N. We will address this later. This estimator is known as the *unbiased sample variance*. S_N^2 denotes the corresponding random variable.

$$S_N^2 = \frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \bar{Y}_N)^2 \tag{3.22}$$

Bias of the unbiased sample variance

As the name suggests, S_N^2 is an unbiased estimator of variance. We show this by proving $E[S_N^2] = \sigma_Y^2$ for all distributions of Y (provided σ_Y^2 is finite).

$$E[S_N^2] = E\left[\frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \bar{Y}_N)^2\right] \tag{3.23}$$

$$= \frac{1}{N-1}E\left[\sum_{i=1}^{N}\left(Y_i^2 - 2\bar{Y}_N\, Y_i + \bar{Y}_N^2\right)\right] \tag{3.24}$$

$$= \frac{1}{N-1}E\left[\sum_{i=1}^{N}Y_i^2 - 2\bar{Y}_N\sum_{i=1}^{N}Y_i + N\,\bar{Y}_N^2\right] \tag{3.25}$$

Using Eq. 3.11, we can replace $\sum_{i=1}^{N} Y_i$ with $N\bar{Y}_N$, and obtain,

$$E[S_N^2] = \frac{1}{N-1}E\left[\sum_{i=1}^{N}Y_i^2 - N\,\bar{Y}_N^2\right] \tag{3.26}$$

$$= \frac{1}{N-1}\left(\sum_{i=1}^{N}E[Y_i^2] - NE[\bar{Y}_N^2]\right) \tag{3.27}$$

Because the Y_i's are identically distributed, we have $E[Y_i^2] = E[Y^2]$:

$$E[S_N^2] = \frac{1}{N-1}\left(NE[Y^2] - NE[\bar{Y}_N^2]\right) \tag{3.28}$$

$$= \frac{N}{N-1}\left(E[Y^2] - E[\bar{Y}_N^2]\right) \tag{3.29}$$

We now apply the identity from Eq. 1.34 to Y and $\bar{Y}_N$:

$$E[Y^2] = \sigma_Y^2 + \mu_Y^2 \tag{3.30}$$

$$E[\bar{Y}_N^2] = \frac{1}{N}\sigma_Y^2 + \mu_Y^2 \tag{3.31}$$

Then,

$$E[S_N^2] = \frac{N}{N-1}\left(\sigma_Y^2 + \mu_Y^2 - \frac{1}{N}\sigma_Y^2 - \mu_Y^2\right) \tag{3.32}$$

$$= \frac{N}{N-1}\left(\frac{N-1}{N}\sigma_Y^2\right) \tag{3.33}$$

$$= \sigma_Y^2 \tag{3.34}$$

□

Variance of the unbiased sample variance

The variance of the unbiased sample variance is a complicated function of the properties of Y. We will not need it in this course, but you can see it derived here [1] for generic Y. A simpler formula can be found if Y is assumed to be Gaussian. In this case the sample variance is distributed as a χ^2 ("chi squared") distribution [2]. This is also beyond the scope of this course.

Biased sample variance

We can now understand now why $N-1$ was used instead of N in Equation 3.21: it lead to an unbiased estimator. However, we will see that there are reasons to prefer the alternative, which is known as the *biased sample variance*.

$$\tilde{S}_N^2 = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{Y}_N)^2 \tag{3.35}$$

The bias of $\tilde{S}_N^2$ can be shown to be $-\frac{1}{N}\sigma_Y^2$, by the same procedure as before (you should check this!). $\tilde{S}_N^2$ is certainly worse than S_N^2 in terms of the bias. What about in terms of variance? Even without an explicit formula for variance, we can prove that it is better. Since $\tilde{S}_N^2 = \frac{N-1}{N}S_N^2$, it follows that $Var[\tilde{S}_N^2] = \left(\frac{N-1}{N}\right)^2 Var[S_N^2]$, and therefore $Var[\tilde{S}_N^2] < Var[S_N^2]$.

We've now seen three examples of estimators: the sample mean, the unbiased sample variance, and the biased sample variance. With this last pair we found that the bias and variance do not always agree in their assessment. But it is not clear which one should be preferred. In archery, is it better to produce a wide cloud of shots that is perfectly centered on the bullseye (zero bias, high variance), or a small cloud that is a little bit off center (small bias, small variance)? The answer is that it depends on how the scores are marked on the target. That is, on how the *error* is measured.

A more systematic approach to quantifying estimator performance is therefore to base it on the expected value of its error: $E\left[\text{Error}(\hat{\Theta}_N, \theta)\right]$, where $\text{Error}(\hat{\Theta}_N, \theta)$ is some measure of distance from θ, which we choose. Using the 1-norm (absolute value) we get the mean absolute error criterion, or MAE. With the 2-norm we get the mean squared error criterion, or MSE. Both are useful. In this course we will focus on the MSE because a) it is more widely used, b) it results in a smooth optimization problem (important for gradient descent), and c) it has interesting analytical properties (bias variance decomposition).

3.1.4 Mean squared error (MSE)

The MSE of an estimator $\hat{\Theta}_N$ is defined as,

$$\text{MSE}\ [\hat{\Theta}_N] = E[(\hat{\Theta}_N - \theta)^2] \tag{3.36}$$

Figure 3.3 shows an illustration. The MSE is the expected value of the red distribution that is drawn along the vertical axis in the figure. This distribution is obtained by mapping samples of $\hat{\Theta}_N$ through a parabola centered at θ, shown in the figure as a red line. Note that, if $\hat{\Theta}_N$ were unbiased and its variance equal to zero, then the MSE would be zero. Note also that one cannot generally estimate the MSE from the data because it depends on θ.

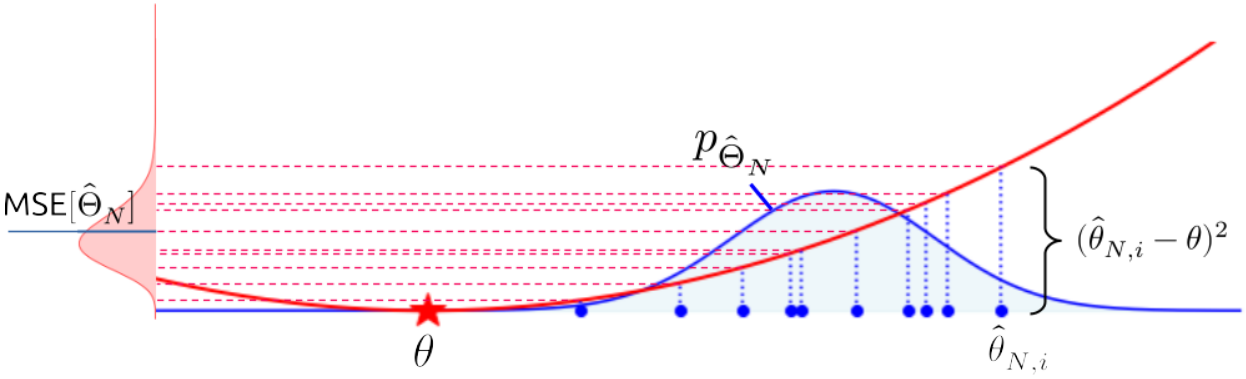

Figure 3.3: MSE of $\hat{\Theta}_N$: See `demo_MSE.py`

The intuition behind the MSE is clear: it measures the expected size of the errors using a Euclidean metric (circles on the archery target). What is less obvious is that the MSE can actually be expressed as a function of the bias and the variance of the estimator. This result is know as the bias-variance decomposition of MSE.

Bias-variance decomposition of MSE

The MSE of an estimator equals the sum of its variance and its bias squared. This is true regardless of the properties of Y and of the value of θ.

$$\text{MSE}[\hat{\Theta}_N] = Var[\hat{\Theta}_N] + \Big(\text{Bias}[\hat{\Theta}_N]\Big)^2 \tag{3.37}$$

Proof: In the definition of MSE (Eq. 3.36) we can add and subtract $E[\hat{\Theta}_N]$ without altering the result.

$$\text{MSE}[\hat{\Theta}_N] = E\left[\Big(\hat{\Theta}_N - E[\hat{\Theta}_N] + E[\hat{\Theta}_N] - \theta\Big)^2\right] \tag{3.38}$$

Expanding the square and applying the linearity of the expectation we get:

$$\text{MSE}[\hat{\Theta}_N] = E\left[\Big(\hat{\Theta}_N - E[\hat{\Theta}_N]\Big)^2\right] + 2E\left[\Big(\hat{\Theta}_N - E[\hat{\Theta}_N]\Big)\Big(E[\hat{\Theta}_N] - \theta\Big)\right] + E\left[\Big(E[\hat{\Theta}_N] - \theta\Big)^2\right] \tag{3.39}$$

Let's look closely at each of these three terms. The first is the variance of $\hat{\Theta}_N$. In the last term, both $E[\hat{\Theta}_N]$ and θ are deterministic (non-random), so the outer expectation can be removed, and the term equals the square of the bias (Eq. 3.7). In the middle term, because $E[\hat{\Theta}_N] - \theta$ is deterministic, it can again be extracted from the expectation. The middle term then becomes:

$$2\Big(E[\hat{\Theta}_N] - \theta\Big)\, E\left[\hat{\Theta}_N - E[\hat{\Theta}_N]\right] \tag{3.40}$$

Again using the linearity of the expectation, this becomes:

$$2\Big(E[\hat{\Theta}_N] - \theta\Big)\Big(E[\hat{\Theta}_N] - E[\hat{\Theta}_N]\Big) \tag{3.41}$$

which equals zero. □

This decomposition gives us a different perspective on the MSE: it balances variance and bias by giving equal weight to both. The bias is squared so that the units match. Next we compute the MSE for our three estimators.

1. MSE of the sample mean:

$$\text{MSE}[\bar{Y}_N] = Var[\bar{Y}_N] + \big(\text{Bias}[\bar{Y}_N]\big)^2 \tag{3.42}$$

$$= \frac{\sigma_Y^2}{N} + 0 \tag{3.43}$$

$$= \frac{\sigma_Y^2}{N} \tag{3.44}$$

   It can be shown that this is the lowest value of MSE achievable by any unbiased estimator of the mean (provided that Y has finite mean and variance). The terminology for this is that the sample mean is the *minimum variance unbiased estimator* (MVUE)

for the mean.

2. MSE of the unbiased sample variance:

$$\text{MSE}[S_N^2] = Var[S_N^2] + \left(\text{Bias}[S_N^2]\right)^2 \tag{3.45}$$
$$= Var[S_N^2] \tag{3.46}$$

Recall that $Var[S_N^2]$ is a complicated function of the properties of Y.

3. MSE of the biased sample variance:

$$\text{MSE}[\tilde{S}_N^2] = Var[\tilde{S}_N^2] + \left(\text{Bias}[\tilde{S}_N^2]\right)^2 \tag{3.47}$$
$$= \left(\frac{N-1}{N}\right)^2 Var[S_N^2] + \left(\frac{\sigma_Y^2}{N}\right)^2 \tag{3.48}$$
$$= \left(\frac{N-1}{N}\right)^2 \text{MSE}[S_N^2] + \left(\frac{\sigma_Y^2}{N}\right)^2 \tag{3.49}$$

Knowing N and σ_Y^2 therefore allows us to decide which of the two estimators fo variance produce a smaller MSE.

3.1.5 Asymptotic properties

We have described several properties of estimators that can be used to assess their performance: bias, variance, and MSE. This has thus far been done assuming a fixed value of N. However, we are also interested in knowing how an estimator performs on datasets of different sizes. A 'good' estimator should naturally benefit from receiving additional data, and therefore its bias and variance should improve as N increases.

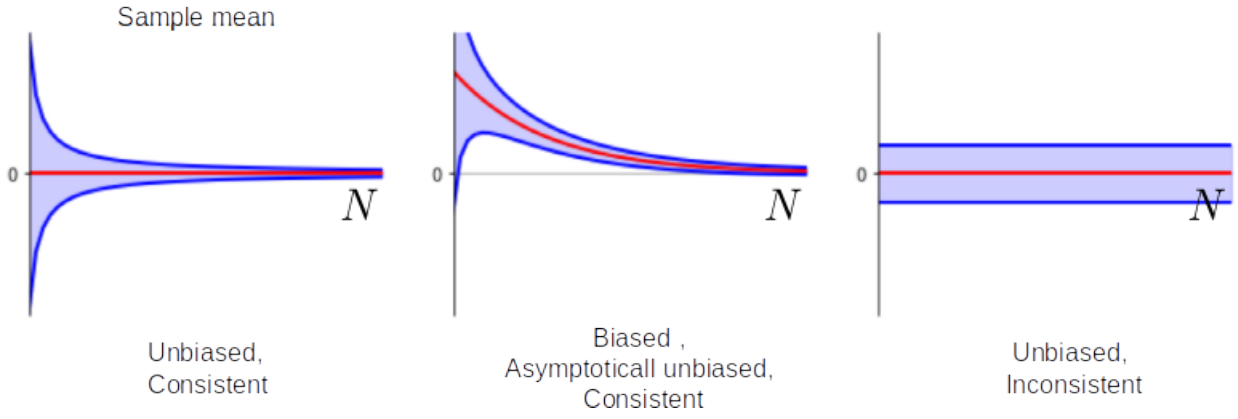

Figure 3.4: Bias ± variance as a function of N for three estimators.

Figure 3.4 illustrates the concepts. Now, instead of considering, say, the sample mean as a single estimator, we think of it as a *sequence* of estimators, with N varying from 1 to

infinity. The figure shows the bias (red) with a variance-sized envelope (blue), as functions of N for three different different estimators. The plot on the left corresponds to the sample mean. As we have seen, the sample mean is unbiased and its variance decays as $1/N$, which implies that it converges to zero as $N \to \infty$.

Asymptotic unbiasedness

An estimator is *asymptotically unbiased* when its bias converges to zero as N increases.

$$\lim_{N\to\infty} \text{Bias}[\hat{\Theta}_N] = 0 \tag{3.50}$$

An estimator that lacks asymptotic unbiasedness will have a persistent error that is immune to data. The left and right plots in Figure 3.4 are unbiased, and therefore asymptotically unbiased. The middle plot is biased for all N, but asymptotically unbiased.

Consistency

A *consistent* estimator is one whose estimates are guaranteed to approach the true value as N gets large. That is,

$$\hat{\Theta}_N \xrightarrow{p} \theta \quad \text{as} \quad N \to \infty \tag{3.51}$$

Here, the '$\xrightarrow{p}$' stands for "converges in probability". The standard notion of convergence ($\to$) does not apply in this situation, since the objects on the left and right side of the symbol are of different types: $\hat{\Theta}_N$ is a random variable and θ is a number. We must specify the sense in which a sequence of random variables is said to "converge" to a number.

Definition. The sequence of random variables $\hat{\Theta}_1, \hat{\Theta}_2, \ldots$ converges *in probability* to θ if, for every $\epsilon > 0$,

$$\lim_{N\to\infty} P\left(|\hat{\Theta}_N - \theta| \geq \epsilon\right) = 0 \tag{3.52}$$

That is, as N gets large, the probability of seeing an estimate with an error greater than ϵ (some small number) converges to zero. And this is true no matter how small we choose ϵ.

How does consistency relate to bias, variance, and MSE? Do these three quantity necessarily tend to zero when the estimator is consistent? The answer to this question is, strictly speaking, "no". There are corner cases of estimators that are consistent, but whose MSE does not tend to zero. However these are contrived. For most real-world distributions, if an estimator is consistent, then its MSE converges to zero (and therefore its bias and variance as well). The converse is *always* true: If MSE converges to zero, then the estimator is consistent. In short, MSE convergence is a slightly stronger property than consistency.

Asymptotic properties of the sample mean

The sample mean, in addition to being unbiased (and therefore asymptotically unbiased), is also a consistent estimator of the mean. Generally it can be difficult to show that an estimator is consistent. However *for unbiased estimators*, the condition for consistency reduces to the

variance tending to zero as $N \to \infty$. This is certainly true of the sample mean:

$$\lim_{N\to\infty} Var[\bar{Y}_N] = \lim_{N\to\infty} \frac{Var[Y]}{N} = 0 \tag{3.53}$$

The sample mean is in fact the best possible estimator of the mean for many processes Y. It is also the reasonable choice if we have no knowledge of the the shape of p_Y. However, there are cases in which other estimators do better. For example, if Y is known to follow a Laplace distribution [3], then the *sample median* is also unbiased and consistent, and has a smaller variance than the sample mean.

This concludes our review of performance criteria for point estimators. We have seen bias, variance, and MSE for fixed N, and two asymptotic criteria: asymptotic unbiasedness and consistency. Next we will see how optimization theory can be put to the task of computing parameter estimates.

3.1.6 Maximum likelihood estimation (MLE)

The three estimators introduced in this chapter were given as raw formulas (Eqs. 3.11, 3.22, and 3.35), with no indication as to how they were obtained. This will now be remedied with the introduction of the maximum likelihood method. MLE is a technique based on optimization theory for generating estimators. As we will see, the MLE approach can be used to produce estimates, not only of mean and variance, but of any properties or parameters of Y. We will also see that estimators generated with MLE enjoy some beneficial asymptotic properties.

We begin by assuming that Y is a member of some parameterized family of distributions, with parameter vector $\underline{\theta}$. Notice that until now we have only considered the point estimation of single parameter θ (the mean or the variance). Maximum likelihood on the other hand, allows the estimation of the entire $\underline{\theta}$ vector at once. Notice also that in deriving the bias and variance of previous estimators we did not make any assumptions about the *family* of Y. For example, the sample mean is unbiased for all distributions of Y (parameterized or otherwise), provided only that μ_Y is finite. The MLE approach however requires that we begin by specifying a parametric family for Y.

The problem is cast as a search over the space of possible $\underline{\theta}$'s. We seek the parameter setting (i.e. the member of the parametric family) that maximizes an objective function that can be computed from the data. The maximum likelihood method uses the *likelihood* of the parameter setting, given the observed data. That is, for each possible setting for $\underline{\theta}$, it computes the probability (a.k.a. the likelihood) of obtaining the data that we have observed, assuming that Y equals the distribution corresponding to that setting. The likelihood of a candidate parameter setting $\hat{\underline{\theta}}$ is computed as follows:

$$\mathcal{L}(\hat{\underline{\theta}}\,;\mathcal{D}) \;=\; \prod_{i=1}^{N} p_Y(y_i;\hat{\underline{\theta}}) \tag{3.54}$$

The likelihood $\mathcal{L}$ of parameter setting $\hat{\underline{\theta}}$, given the fixed dataset $\mathcal{D} = \{y_i\}_N$, is the product of probabilities of the individual samples. This follows from the iid assumption of the samples. The maximum likelihood estimate is the parameter setting $\hat{\underline{\theta}}_{\mathsf{MLE}}$ that maximizes the likelihood function. This gives us the model $p_Y(y\,; \hat{\underline{\theta}}_{\mathsf{MLE}})$ that best fits the observed data. Here is the statement of the problem as an optimization problem:

Given $\mathcal{D} = \{y_i\}_N$

$$\hat{\underline{\theta}}_{\mathsf{MLE}} = \underset{\hat{\underline{\theta}}}{\operatorname{argmax}} \prod_{i=1}^{N} p_Y(y_i; \hat{\underline{\theta}}) \tag{3.55}$$

Example 3.1.1. We are presented with a bag containing 4 marbles. Each marble is either black or white. We are allowed to extract and look at a marble 5 times, each time returning the marble to the bag. Upon doing this, we observe the following sequence: {black, white, black, white, white}. Find the maximum likelihood estimate of the number of black marbles in the bag.

Solution. We begin the MLE procedure by proposing a distribution family for our sampling process. Since the bag produces only two possible outcomes (black and white), the reasonable choice is a Bernoulli distribution, with parameter α for the probability of drawing a black marble. The quantity that we are interested in estimating – the number θ of black marbles in the bag – relates to α with $\theta = 4\alpha$. For example, if there are two black marbles ($\theta = 2$), the probability of drawing one is $\alpha = {}^{2}\!/_{4} = {}^{1}\!/_{2}$. Extracting a marble from the bag is then like sampling from a Bernoulli distribution $Y \sim \mathcal{B}({}^{\theta}\!/_{4})$

$$p_Y(y; \theta) = \begin{cases} {}^{\theta}\!/_{4} & y \text{ is black} \\ 1 - {}^{\theta}\!/_{4} & y \text{ is white} \end{cases} \tag{3.56}$$

The maximum likelihood estimate is the specific value of $\theta \in \{0, 1, 2, 3, 4\}$ that maximizes the likelihood given the observed dataset $\mathcal{D} = \{1, 0, 1, 0, 0\}$. Here we have encoded black marbles with 1 and white marbles with 0. Next we plug the Bernoulli pdf into the likelihood function and evaluate it using the observed dataset.

$$\begin{aligned} \mathcal{L}(\hat{\theta}; \mathcal{D}) &= \prod_{i=1}^{N} p_Y(y_i; \hat{\theta}) && (3.57) \\ &= p_Y(1; \hat{\theta})\, p_Y(0; \hat{\theta})\, p_Y(1; \hat{\theta})\, p_Y(0; \hat{\theta})\, p_Y(0; \hat{\theta}) && (3.58) \\ &= \left({}^{\hat{\theta}}\!/_{4}\right)^2 \left(1 - {}^{\hat{\theta}}\!/_{4}\right)^3 && (3.59) \end{aligned}$$

The easiest way to find the maximum of this function is to evaluate it on each of the five possible values for $\hat{\theta} \in \{0, 1, 2, 3, 4\}$, and choose the largest value. Of these numbers, $({}^{1}\!/_{2})^2\,({}^{1}\!/_{2})^3 = {}^{1}\!/_{32}$ is the largest. Therefore $\hat{\underline{\theta}}_{\mathsf{MLE}} = 2$

Returning to the general problem of Equation 3.55, it is often convenient to apply the logarithm function to the objective. This can make the problem more tractable, both analyt-

$\hat{\theta}$	0	1	2	3	4
$\mathcal{L}(\hat{\theta};\mathcal{D})$	0	$(1/4)^2\,(3/4)^3$	$(1/2)^2\,(1/2)^3$	$(3/4)^2\,(1/4)^3$	0

ically and numerically, and it does not alter the result since the logarithm is a monotonically increasing function. The problem then becomes:

$$\underline{\hat{\theta}}_{\mathsf{MLE}} = \underset{\underline{\hat{\theta}}}{\text{argmax}}\ \ln \mathcal{L}(\underline{\hat{\theta}};\mathcal{D}) \tag{3.60}$$

$$= \underset{\underline{\hat{\theta}}}{\text{argmax}}\ \ln\left(\prod_{i=1}^{N} p_Y(y_i;\underline{\hat{\theta}})\right) \tag{3.61}$$

$$= \underset{\underline{\hat{\theta}}}{\text{argmax}}\ \sum_{i=1}^{N} \ln p_Y(y_i;\underline{\hat{\theta}}) \tag{3.62}$$

The difficulty of solving this problem depends on the parametric family that is assumed. Next we will use the MLE technique to derive estimators of the mean μ_Y and the variance σ_Y^2 of a Gaussian random variable. Previously we had found the sample mean $\bar{Y}_N$ and the biased sample variance $\tilde{S}_N^2$ to be good estimators for these quantities. We will now see how these can be derived (for Gaussian data) using maximum likelihood.

MLE with Gaussian data

With $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$, the pdf of Y is,

$$p_Y(y;\hat{\mu},\hat{\sigma}^2) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{1}{2}\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2}\right) \tag{3.63}$$

The log-likelihood is then:

$$\ln \mathcal{L}(\hat{\mu},\hat{\sigma}^2;\mathcal{D}) = \sum_{i=1}^{N} \ln p_Y(y_i;\hat{\mu},\hat{\sigma}^2) \tag{3.64}$$

$$= \sum_{i=1}^{N} \ln\left(\frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{1}{2}\frac{(y-\hat{\mu})^2}{\hat{\sigma}^2}\right)\right) \tag{3.65}$$

$$= -\frac{N}{2}\ln(2\pi\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{N}(y_i-\hat{\mu})^2 \tag{3.66}$$

We convert the problem into a minimization problem by flipping the sign of the objective.

$$\left(\hat{\mu}_{\mathsf{MLE}}, \hat{\sigma}^2_{\mathsf{MLE}}\right) = \underset{\hat{\mu},\hat{\sigma}^2}{\text{argmin}}\ \left(\frac{N}{2}\ln(2\pi\hat{\sigma}^2) + \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{N}(y_i-\hat{\mu})^2\right) \tag{3.67}$$

The cost function of this problem is the negative of the log-likelihood:

$$J(\hat{\mu}, \hat{\sigma}^2) = \frac{N}{2} \ln(2\pi\hat{\sigma}^2) + \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{N} (y_i - \hat{\mu})^2 \tag{3.68}$$

The only constraint on the parameters is $\hat{\sigma}^2 > 0$. Hence, the feasible set is open (i.e. all points are interior points). Because J is differentiable everywhere in the feasible set, the first order condition tells us that all global solutions (if any exist) are to be found amongst the stationary points of J. Next we find these stationary points by equating each of the partial derivatives to zero. We begin with the partial derivative with respect to $\hat{\mu}$.

$$\frac{\partial J}{\partial \hat{\mu}} = \frac{\partial}{\partial \hat{\mu}} \left(\frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{N} (y_i - \hat{\mu})^2 \right) \tag{3.69}$$

$$= \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{N} \frac{\partial}{\partial \hat{\mu}} (y_i - \hat{\mu})^2 \tag{3.70}$$

$$= -\frac{1}{\hat{\sigma}^2} \sum_{i=1}^{N} (y_i - \hat{\mu}) \tag{3.71}$$

$$= \frac{N\hat{\mu}}{\hat{\sigma}^2} - \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{N} y_i \tag{3.72}$$

Equating this to zero we find that the maximum likelihood estimate of the mean for a Gaussian variable is the sample mean.

$$\hat{\mu}_{\mathsf{MLE}} = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{3.73}$$

Next we take a partial derivative with respect to $\hat{\sigma}^2$.

$$\frac{\partial J}{\partial \hat{\sigma}^2} = \frac{\partial}{\partial \hat{\sigma}^2} \left(\frac{N}{2} \ln(2\pi\hat{\sigma}^2) + \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{N} (y_i - \hat{\mu})^2 \right) \tag{3.74}$$

$$= \frac{N}{2\hat{\sigma}^2} - \frac{1}{2\hat{\sigma}^4} \sum_{i=1}^{N} (y_i - \hat{\mu})^2 \tag{3.75}$$

$$= \frac{1}{2\hat{\sigma}^2} \left(N - \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{N} (y_i - \hat{\mu})^2 \right) \tag{3.76}$$

Equating this to zero we find that the maximum likelihood estimate of the variance of a Gaussian variable is the biased sample variance.

$$\hat{\sigma}^2_{\mathsf{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{\mu}_N)^2 \tag{3.77}$$

Properties of maximum likelihood estimators

The maximum likelihood optimization problem is a straightforward technique for finding the member of a family of pdfs that best fits a given dataset. In posing this problem, we made no reference to the desirable properties of unbiasedness, low MSE, or consistency. So it is reasonable to ask whether the MLE achieves any of these properties. Here are some facts.

1. In general, maximum likelihood estimators have no finite-sample properties. For each fixed N, it is not necessarily unbiased, and it doesn't necessarily achieve minimum MSE. However we know of at least one case where it does have these properties: the sample mean of a Gaussian variable.

2. Maximum likelihood estimators have good asymptotic properties. They are consistent and, setting aside a few corner cases, they are asymptotically unbiased.

3.2 Confidence intervals

While point estimates provide single estimates of the value of a parameter θ, confidence intervals go further and deliver an *interval* that is presumed to contain θ with a given level of certainty. The formulation of the confidence interval problem is as follows.

Given:

1. a dataset: $\mathcal{D} = \{y_i\}_N \overset{\text{iid}}{\sim} Y$,

2. that Y is a member of a parameterized family of distributions: $Y \sim p_Y(y;\theta)$,

3. a desired *confidence level* $\gamma \in [0,1]$,

find a *confidence interval* $I_{N,\gamma}$ that contains the parameter θ with confidence γ. We have indexed the interval with N and γ in order to track its dependence on both the size of the dataset and the confidence level. Also notice that we have made an assumption about the parameterized family of Y. With point estimates we did not necessarily have to do this - the sample mean and variance estimators of sections 3.1.2 and 3.1.3 are valid for any Y (provided μ_Y and σ^2_Y are finite). Here though, we must specify a distribution family for Y since this will affect the width of the interval.

The confidence interval $I_{N,\gamma} = [A, B]$ is computed as a function of the random data, and hence its limits A and B are themselves random variables. $I_{N,\gamma}$ can therefore be thought of as a *random interval*. The *confidence level* γ is the probability that $I_{N,\gamma}$ will contain θ.

$$\gamma = P(\theta \in I_{N,\gamma}) \tag{3.78}$$

In this expression, θ is the deterministic but unknown true value of the parameter, and $I_{N,\gamma}$ is random. This is illustrated in Figure 3.5. The parameter being estimated in the figure is the mean of a normal distribution. The distribution is shown at the top and its true mean μ is indicated with a vertical dashed line. Each row shows a different confidence interval computed with a different iid dataset. The data are the gray dots. The intervals are centered on the sample means, and their widths are also functions of the data. Red intervals are ones which do not contain the true mean, blue intervals are ones that do. The confidence level γ is the probability that the interval will contain the parameter. In this case it is 0.5 – about half of the intervals are red, and half are blue. Of course, in real applications we cannot know whether θ lies in $I_{N,\gamma}$ because we do not know θ. However we do know that using a large γ will increase the probability that θ lies in $I_{N,\gamma}$, because it will result in a larger interval.

The difficulty of computing a confidence interval depends on the parameter being estimated, and on the distribution family that has been assumed. We begin with the simplest case: the mean of a normal distribution.

3.2.1 Confidence interval for the mean of a normal distribution

Given a dataset $\mathcal{D} = \{y_i\}_N$ that has been iid sampled from a normal distribution $\mathcal{N}(\mu_Y, \sigma_Y^2)$, we wish to produce an interval $I_{N,\gamma}$ that contains μ_Y with confidence γ.

The first thing to notice is that, because the normal distribution is symmetric, it is reasonable to center the confidence interval on an unbiased point estimate of the mean. The sample mean $\hat{\mu}_N$ is the natural choice. Then $I_{N,\gamma}$ can then be expressed as,

$$I_{N,\gamma} = [\hat{\mu}_N - \rho, \hat{\mu}_N + \rho] \tag{3.79}$$

or equivalently, $I_{N,\gamma} = \hat{\mu}_N \pm \rho$, where ρ is the *radius* of the interval. We can now replace symbols in Eq. 3.78: $\theta \leftarrow \mu_Y$, $A \leftarrow \bar{Y}_N - \rho$, $B \leftarrow \bar{Y}_N + \rho$. This produces an alternative expression for the confidence level γ.

$$\gamma = P(\mu_Y \in [\hat{\mu}_N - \rho, \hat{\mu}_N + \rho]) = P\left(|\mu_Y - \bar{Y}_N| < \rho \right) \tag{3.80}$$

The confidence level is then the probability that the true mean μ_Y is within ρ of the sample mean, or equivalently the probability that the sample mean is within ρ of the true mean. But we know the distribution of the sample mean. We have seen that it is unbiased, that

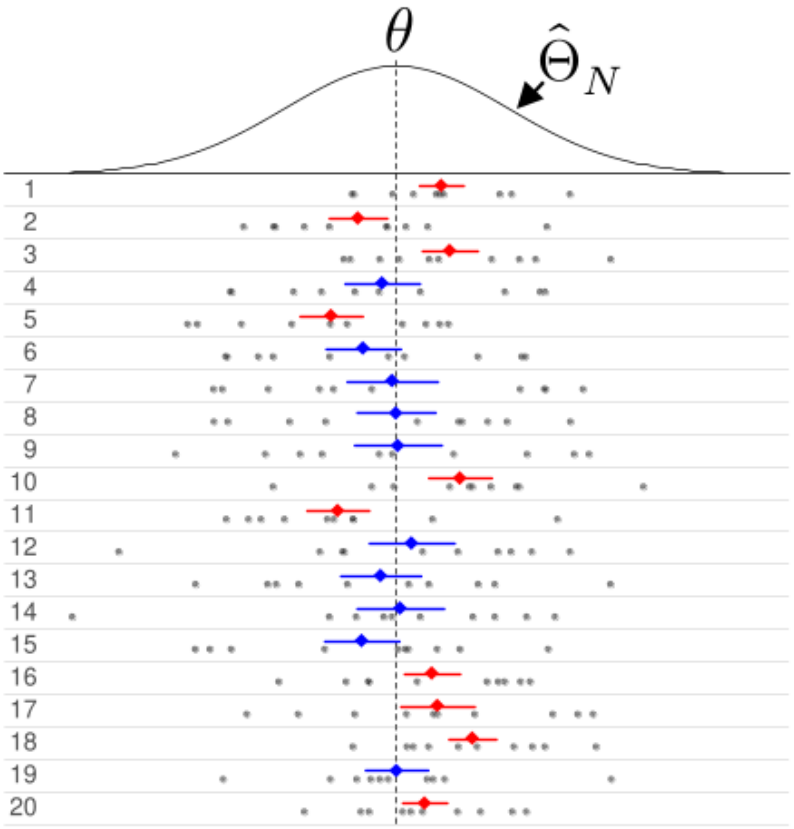

Figure 3.5: Several confidence intervals for the mean of a normal distribution (Source: Wikipedia)

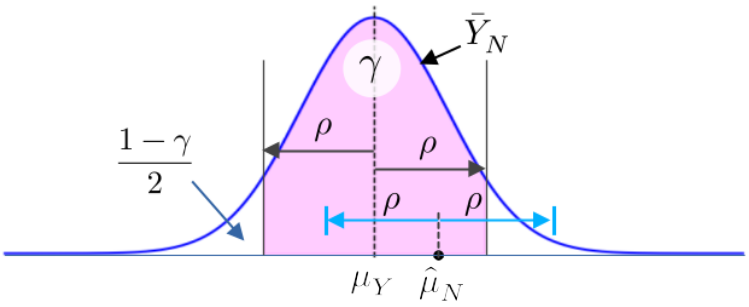

Figure 3.6: Distribution of the sample mean and computation of ρ

its variance is inversely proportional to N, and that it is normal when Y is normal.

$$\bar{Y}_N \sim \mathcal{N}\left(\mu_Y, \frac{\sigma_Y^2}{N}\right) \tag{3.81}$$

Figure 3.6 shows the distribution of $\bar{Y}_N$, along with a confidence interval centered on $\hat{\mu}_N$ (a sample of $\bar{Y}_N$) and with radius ρ. The radius ρ is defined to capture an area of γ underneath $p_{\bar{Y}_N}$. This can be done prior to collecting the data (assuming σ_Y^2/N is known). Then, once $\hat{\mu}_N$ is obtained, this radius can be used to construct the confidence interval centered at $\hat{\mu}_N$.

It will actually be more convenient (for the purpose of using lookup tables) to use a normalized version of $\bar{Y}_N$. By shifting $\bar{Y}_N$ to the origin and scaling by its standard deviation, we obtain a new quantity called Z, which is distributed as a unit normal.

$$Z = \frac{\bar{Y}_N - \mu_Y}{\sigma_Y/\sqrt{N}} \sim \mathcal{N}(0,1) \tag{3.82}$$

In terms of Z, the confidence level is the probability that Z is within $\bar{\rho} = \left(\sqrt{N}/\sigma_Y\right)\rho$ of 0.

$$\gamma = P\left(\,|Z| < \bar{\rho}\,\right) \tag{3.83}$$

We can use the cdf of Z (i.e. the unit normal cdf) to compute this probability. We denote the unit normal cdf with $\Phi_{\mathcal{N}}$. Then,

$$\gamma \;=\; P\left(\,|Z| < \bar{\rho}\,\right) \;=\; \Phi_{\mathcal{N}}(\bar{\rho}) - \Phi_{\mathcal{N}}(-\bar{\rho}) \tag{3.84}$$

By symmetry of the normal distribution we have that $\Phi_{\mathcal{N}}(\bar{\rho}) = 1 - \Phi_{\mathcal{N}}(-\bar{\rho})$. Therefore,

$$\gamma \;=\; P\left(\,|Z| < \bar{\rho}\,\right) \;=\; 1 - 2\Phi_{\mathcal{N}}(-\bar{\rho}) \tag{3.85}$$

From this we obtain an expression for $\bar{\rho}$ in terms of γ,

$$\bar{\rho} = -\Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right) \tag{3.86}$$

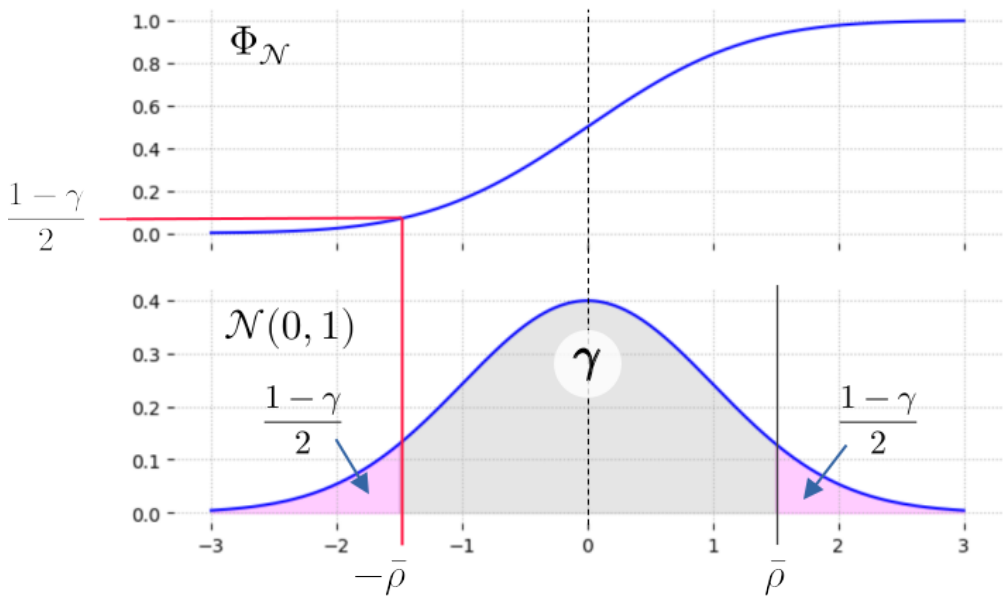

Figure 3.7: Radius of the confidence interval.

Figure 3.7 illustrates the calculation. The area underneath the unit normal pdf is split into three parts. The middle part has area γ, the two sides have area $(1-\gamma)/2$. The inverse cdf evaluated at $(1-\gamma)/2$ is a negative number. We can eliminate the negative sign by taking the absolute value. The radius ρ of the confidence interval is then found by un-scaling $\bar{\rho}$ (i.e. multiplying by $\sigma_Y/\sqrt{N}$).

$$\rho = \frac{\sigma_Y}{\sqrt{N}} \left| \Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right) \right| \tag{3.87}$$

The value of $\Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right)$ can be found using a lookup table such as the one in the appendix, or with Python. This is demonstrated next.

Example 3.2.1. A sample of 10 resistors is taken from a process for manufacturing 200 ohm resistors. The sample has a mean of 195 ohms. Construct a 95% confidence interval for the mean assuming that the true distribution is Gaussian, with a standard deviation of 7 ohms.

Solution

The problem statement specifies $N = 10$, $\hat{\mu}_N = 195$, $Y \sim \mathcal{N}(\mu_Y, 7^2)$, and $\gamma = 0.95$. The confidence interval is centered at $\hat{\mu}_N$, and its radius is found with Equation 3.87. Use the lookup table [4] to find,

$$\Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right) = \Phi_{\mathcal{N}}^{-1}\left(0.025\right) = -1.96 \tag{3.88}$$

This can be done in Python with `scipy.stats.norm().ppf(0.025)`.

The radius of the interval is then:

$$\rho = \frac{7}{\sqrt{10}}(1.96) = 4.34 \tag{3.89}$$

The confidence interval is therefore 195 ± 4.34 ohms or $[190.66, 199.34]$ ohm. □

We can see from Equation 3.87 that the width of the confidence interval decreases as the size of the sample grows. This reflects the consistency of the sample mean estimator, which delivers more precise estimates when given more data. Conversely, a larger process variance σ_Y^2 results in a larger confidence interval. We can be less sure of the location of the mean when the data is noisy. Finally, because the cdf of the normal distribution $\Phi_{\mathcal{N}}$ is an increasing function, it follows that its inverse is also increasing, which means that $\Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right)$ is a decreasing function of γ. But since $\Phi_{\mathcal{N}}^{-1}\left(\frac{1-\gamma}{2}\right)$ is negative on $\gamma \in (0,1)$, its absolute value is an *increasing* function of γ. So, as γ increases, the width of the confidence interval also increases. Our confidence that the mean is contained in the interval grows as the interval gets larger.

Confidence interval for the mean: Unknown σ_Y

A significant limitation of the approach we've just outlined is that it requires a-priori knowledge of σ_Y, the standard deviation of Y. This is usually not known. The best we can typically do is to estimate σ_Y, for example by taking the square root of the unbiased sample variance:

$$\hat{\sigma}_N = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \hat{\mu}_N)^2} \tag{3.90}$$

This works well, and the resulting estimate can be used in Equation 3.87 in place of σ_Y whenever N is sufficiently large. While the threshold for large N will depend on the underlying distribution, a good rule of thumb which we will adopt in this course is $N > 30$.

If, however, the sample size is small ($N \leq 30$), then using $\hat{\sigma}_N$ in place of σ_Y can induce a large error in the confidence interval. This problem was solved in the early 1900's for the Gaussian case by William Sealy Gosset, who published the result under the pseudonym of "Student". Gosset derived an expression for the distribution of *t statistic*, which is obtained by replacing the true standard deviation with the square root of the unbiased sample variance in Equation 3.82.

$$t = \frac{\bar{Y}_N - \mu_Y}{\sqrt{S_N^2/N}} \sim t(\nu) \tag{3.91}$$

The notation is a little bit confusing. The t on the left is the statistic (a random variable), which is analogous to Z in Eq. 3.82. $t(\nu)$ on the right refers to the t distribution, which is a family of distributions parameterized by ν ("nu"), the number of *degrees of freedom*. For univariate problems such as the ones we will be doing, $\nu = N-1$. Figure 3.8 shows examples of the t distribution for different values of ν. As ν grows, the distribution converges to a

normal distribution. For smaller ν (small datasets), the variance of the distribution is larger, due to the larger uncertainty in the estimate of σ_Y.

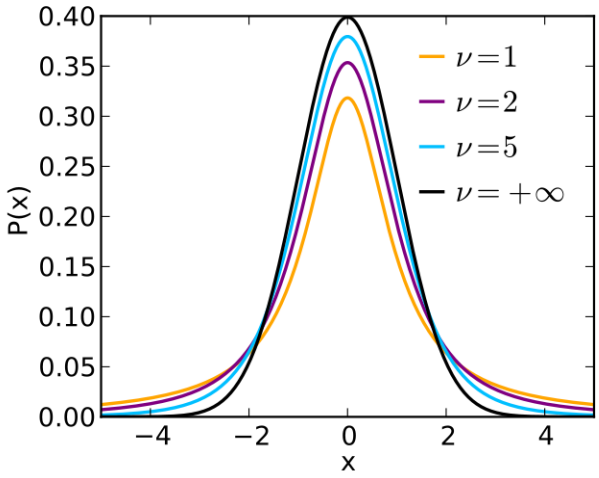

Figure 3.8: Student's t-distribution

The procedure for computing confidence intervals with unknown σ_Y is identical to the case when σ_Y is known, except that we use a lookup table for the inverse cdf of the t-distribution instead of the normal distribution.

$$\rho = \frac{\hat{\sigma}_N}{\sqrt{N}} \left| \Phi^{-1}_{t(\nu)} \left(\frac{1-\gamma}{2} \right) \right| \tag{3.92}$$

Example 3.2.2. Repeat Example 3.2.1, but assuming unknown standard deviation.

Solution
This time, the radius of the confidence interval is found with a lookup table for the inverse cdf of the t distribution, using $\nu = N - 1 = 9$ (see [5])

$$\Phi^{-1}_{t(9)}\left(0.025\right) = 2.26 \tag{3.93}$$

This can be done in Python with `scipy.stats.t(df=9).ppf(0.025)`.
The radius of the interval is then:

$$\rho = \frac{\hat{\sigma}_N}{\sqrt{10}} \left(2.26\right) \tag{3.94}$$

At this point we would estimate the standard deviation $\hat{\sigma}_Y$ for the given dataset using Eq. 3.90, and then use the resulting ρ to find the interval. However since no dataset has been given, we will leave it there.

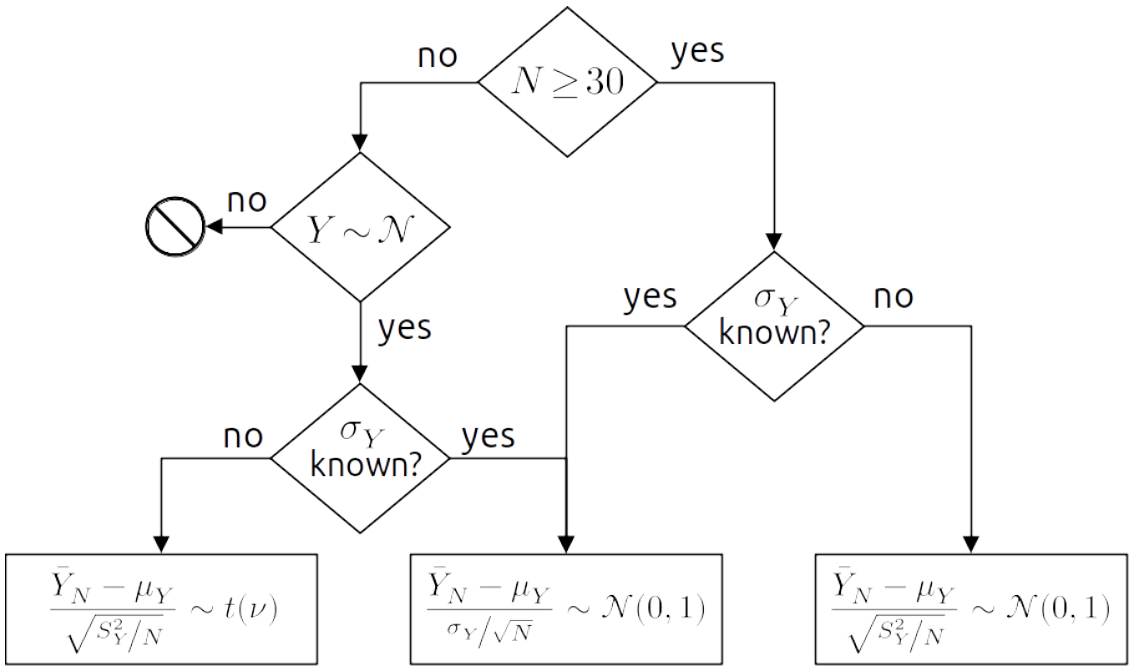

Figure 3.9: Diagram for deciding which distribution to use to compute ρ

Confidence interval of the mean: Non-Gaussian data

So far we have used the assumption of Gaussian Y to assert that the sample mean is Gaussian (Eq. 3.81). We then proceeded to compute the radius of the confidence interval by normalizing the sample mean (Eq. 3.82). Because the normalization factor involves the standard deviation of Y, this generates two cases: known and unknown σ_Y, and two corresponding distributions for the normalized sample mean: $Z \sim \mathcal{N}(0,1)$ and $t \sim t(\nu)$.

If we now drop the assumption of Gaussian Y, then we can no longer be sure that $\bar{Y}_N$ is Gaussian. This does not necessarily mean that it is unknown. There may be non-Gaussian distributions for which the distribution of $\bar{Y}_N$ is known. However when this is not the case, we must appeal to the central limit theorem, which tells us that if N is sufficiently large, then we are justified in approximating $\bar{Y}_N$ as Gaussian.

Here we will again use $N > 30$ as a threshold for invoking the CLT. Having made that assumption, the rest of the procedure is identical. We can normalize the sample mean by subtracting the mean and dividing it by the true or sample standard deviation. If σ_Y is known, the result is distributed as $\mathcal{N}(0,1)$. If σ_Y is unknown and S_N^2 is used instead, then the result is distributed as $t(\nu)$. Except, because N large, ν is also large and $t(\nu)$ becomes indistinguishable from $\mathcal{N}(0,1)$, *so the two cases collapse into one*. Figure 3.9 provides a summary diagram.

Example 3.2.3. The following problem is taken from chapter 5.2 of Navidi.
A soft-drink manufacturer purchases aluminum cans from an outside vendor. A random sample of 70 cans is selected from a large shipment, and each is tested for strength by

applying an increasing load to the side of the can until it punctures. Of the 70 cans, 52 meet the specification for puncture resistance. Find a 95% confidence interval for the proportion of cans in the shipment that meet the specification.

Solution

The can-testing process is a Bernoulli random variable with an unknown probability of success (the can meets the specification) of α. We know about $Y \sim \mathcal{B}(\alpha)$ that,

$$E[Y] = \alpha \tag{3.95}$$
$$Var[Y] = \alpha(1-\alpha) \tag{3.96}$$

Hence our goal is to put a 95% confidence interval on the mean of Y. Our dataset consists of $N = 70$ samples:

$$\mathcal{D} = \{y_i\}_N = \{0, 1, 1, 0, 1, 0, 0, ...\} \tag{3.97}$$

with 52 ones and 18 zeros. The sample mean is then:

$$\hat{\mu}_N \;=\; \frac{1}{N}\sum_{i=1}^{N} y_i \;=\; \frac{52}{70} \;\approx\; 0.74 \tag{3.98}$$

and the sample variance is:

$$\hat{\sigma}_N^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \hat{\mu}_N)^2 \tag{3.99}$$
$$= \frac{1}{69}\left(52 \times \left(1 - \frac{52}{70}\right)^2 + 18 \times \left(\frac{52}{70}\right)^2\right) \tag{3.100}$$
$$= \frac{52 \times 18}{69 \times 70} \tag{3.101}$$
$$\approx 0.194 \tag{3.102}$$

Because N is large, we can assert that the normalized sample mean is t-distributed:

$$t = \frac{\bar{Y}_N - \mu_Y}{\sqrt{S_N^2/N}} \sim t(N-1) \tag{3.103}$$

and furthermore that $t(N-1) \approx \mathcal{N}(0,1)$. Therefore we can use the Gaussian "Z-tables" to find the confidence interval for $\alpha = \mu_Y$.

3.3 Hypothesis tests

One of the important lessons of Bayes' rule is that you can never be entirely certain about anything you learn from data. Our experiments may increase or decrease the credence we give to different statements, but these will never reach 0 or 1 by the data alone. Looking

at Bayes' formula (Eq. 1.99), one may counter that the posterior belief will equal zero if the likelihood of the measurement is zero. However, no observation, no matter how unlikely, can carry zero likelihood. For example, a single measurement of a speed exceeding the speed of light would not invalidate Einstein's special theory of relativity, since it may be due to measurement error. Instead, a series of such measurements might significantly reduce our credence for the theory, to the point of rendering it invalid.

How then can we judge whether a statement based on data should be accepted or rejected? To take an example from manufacturing, consider a specification for a steel production process that requires the average tensile strength of the material to be 250 MPa. To monitor this, three specimens are taken from each batch, their tensile strength is measured, and averaged. Suppose that the average strength of a particular sample of three specimens is 249 MPa. Does this provide sufficient evidence to declare that the batch is unusable? Perhaps this small discrepancy of 1 MPa is due to the inherent uncertainty of the sample mean. If so, we should not conclude that the batch is faulty, but rather we may decide to repeat the test. What threshold should we use for deciding that a batch should be rejected?

Hypothesis tests are used in cases such as this one, where we wish to choose between two competing hypotheses. One of the two is the *null* hypothesis. This is the one that receives the benefit of the doubt. In our example, the null hypothesis claims that the average strength is 250 MPa, and that the observed deviation is due to the natural fluctuations of the sample mean. This is not something that can be strictly disproven. Rather, we can only "reject" the null hypothesis if the data shows that it is very unlikely. If we find that the probability of obtaining a sample mean of 249 MPa when $\mu_Y = 250$ MPa is below a threshold, then we can reject the notion that $\mu_Y = 250$ MPa.

The null hypothesis is denoted with H_0, and the *alternate* hypothesis is H_1. In our example,

$$H_0: \quad \mu_Y = 250 \tag{3.104}$$

$$H_1: \quad \mu_Y < 250 \tag{3.105}$$

Y here is the random variable for the tensile strength. The null hypothesis asserts that the mean of Y is 250 MPa. The alternate hypothesis asserts that it is less than 250 MPa.

The first step in conducting a hypothesis test is to choose the threshold for rejecting the null hypothesis. This is called the *significance level* and it is denoted with α. In the sciences, results that meet significance levels of $\alpha = 0.05$ are considered to be "statistically significant". Results that satisfy $\alpha = 0.01$ are considered "highly significant". If the likelihood of the observed data is less than α under the assumption that H_0 is true, then we reject H_0.

The next step is to choose a *test statistic* with which to quantify the likelihood of the data. The test statistic is a function of the data. Its distribution should be fixed and known under the assumption that H_0 is true. Knowing the distribution of the test statistic, we can evaluate the probability of obtaining the observed data (the p-value), and compare this probability to the significance level.

Let's apply this procedure to our example. We'll assume that the standard deviation of the tensile strength of specimens is known to be $\sigma_Y = 5$ MPa. Then the standard deviation

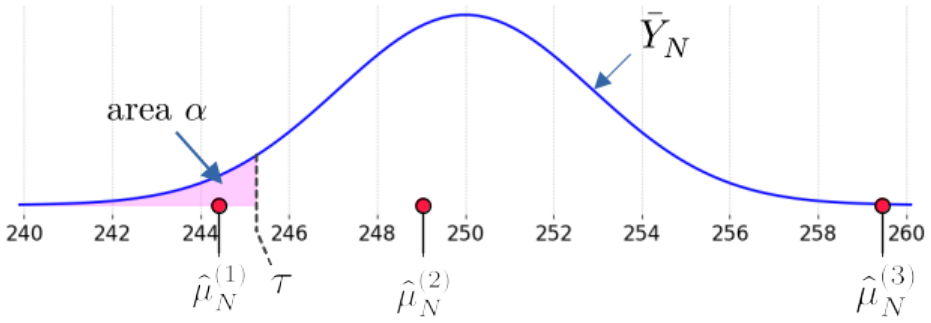

Figure 3.10: Drawing conclusions from a hypothesis test.

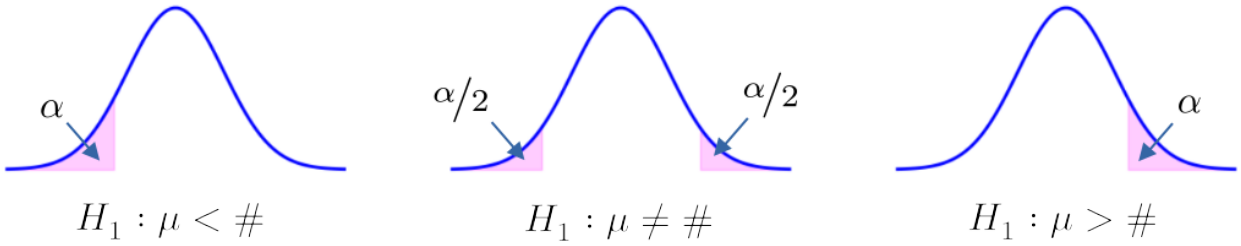

Figure 3.11: Critical regions for three types of alternate hypotheses.

of a sample of 3 specimens is $5/\sqrt{3}$ MPa, since $N = 3$. If in addition we assume that the distribution of tensile strengths is Gaussian, then we can conclude that, under the null hypothesis, $\bar{Y}_3 \sim \mathcal{N}(250, 5/\sqrt{3})$.

The next step is to use the test statistic to evaluate the likelihood of the observed data under the null hypothesis.

We will "reject H_0 in favor of H_1" if the evidence summarized in the test statistic is sufficiently at odds with H_0 and also favors H_1. Figure 3.10 provides an illustration. The figure shows the *critical region* (shaded in pink), which is an interval with probability α that is unlikely under H_0, but compatible with H_1. If the test statistic falls within the critical region, then we reject H_0 and accept H_1. This is the case for $\hat{\mu}_N^{(1)}$ in the figure. $\hat{\mu}_N^{(2)}$ on the other hand, falls outside of the critical region. In this case it is reasonable to assume that the result is due to the natural variations of the sample mean, and therefore it does not support the alternate hypothesis. In this case we "fail to reject H_0 in favor of H_1". $\hat{\mu}_N^{(3)}$ is an example of a very improbable result – even more improbable than $\hat{\mu}_N^{(1)}$. This result supports the notion that H_0 is incorrect. However it does not support the alternate hypothesis that $\mu_Y < 250$. Hence the conclusion is again a "failure to reject H_0 in favor of H_1".

We will consider three types of alternate hypotheses: ones with a "$<$" symbol, ones with a "$\neq$" symbol, and ones with a "$>$" symbol. Figure 3.11 shows the critical regions for each of these three types.

Finally, the rules for selecting a distribution for the radius of a confidence interval (Figure 3.9) also apply to the selection of a statistic for hypothesis tests. This is demonstrated in the following examples.

3.4 Mixture Gaussian Models

We return now to the topic of point estimation to demonstrate the application of maximum likelihood estimation to a more complicated distribution family: the *Gaussian mixture*. A Gaussian mixture is a distribution which, like $p_W(w)$ in Figure 1.15, consists of a weighted sum of Gaussians. Recall Example 1.9.7 with the three vehicle types. Suppose that we only have measurements of the weights W of the vehicles (and not their type V), and we wish to build a model for the joint distribution of weight and type based on this information. In other words, we seek to estimate p_{VW} (Figure 1.14) from samples of p_W (left-hand side of Figure 1.15). How can we do this?

To answer the question, notice that to construct the joint distribution it will suffice to estimate the marginal distribution of V (right-hand side of Figure 1.15) and the conditional distribution of W given V (Figure 1.16). With these we can construct the joint distribution using the definition of the conditional distribution:

$$p_{VW}(\text{scooter}, w) = p(w \mid \text{scooter})\, p_V(\text{scooter}) \tag{3.106}$$
$$p_{VW}(\text{bicycle}, w) = p(w \mid \text{bicycle})\, p_V(\text{bicycle}) \tag{3.107}$$
$$p_{VW}(\text{moped}, w) = p(w \mid \text{moped})\, p_V(\text{moped}) \tag{3.108}$$

Our focus will be on estimating the discrete distribution p_V and the three continuous distributions $p(w \mid \text{scooter})$, $p(w \mid \text{bicycle})$, and $p(w \mid \text{moped})$. We pose this as a point estimation problem, and solve it using the maximum likelihood technique.

Let's first generalize the notation. Instead of W, we'll use Y for the *observed* continuous quantity. Instead of V, we will use Z for the *unobserved class* of the items. There are K of these classes ($K = 3$ in the example), indexed with $k = 1 \ldots K$. The unknown marginal probabilities of the classes ($p_V(v)$ in the example) are denoted with π_k. Then,

$$\sum_{k=1}^{K} \pi_k = 1 \tag{3.109}$$
$$\pi_k \geq 0 \qquad \forall k \in \{1 \ldots K\} \tag{3.110}$$

In this generic notation, our goal is to build a model for the joint distribution $p_{ZY}(k, y)$ for a multivariate random variable (Z, Y), where Z is discrete-valued ($k \in \{1 \ldots K\}$) and Y is continuous-valued, based on observations of Y alone. And we will do this by estimating the class proportions π_k and the conditional distribution $p(y \mid Z = k)$ for each k in $\{1 \ldots K\}$.

To apply the maximum likelihood technique, we must first propose a parametric family for the conditional distribution $p(y \mid Z = k)$. Here we will assume they are Gaussian.

$$p(y \mid Z = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{1}{2}\frac{(y - \mu_k)^2}{\sigma_k^2}\right) = \mathcal{N}_k(y) \tag{3.111}$$

The second equality defines the short-hand $\mathcal{N}_k(y)$ for the Gaussian pdf of the k'th class. Our

problem is thus reduced to the estimation of K proportions π_k and $2K$ parameters μ_k and σ_k^2 ; $3K$ parameters in total.

We can now apply the maximum likelihood machinery to the estimation of the parameter set $\underline{\theta} = \{(\pi_k, \mu_k, \sigma_k^2)\}_K$. Given a dataset $\mathcal{D} = \{y_i\}_N$, the log-likelihood of $\underline{\theta}$ is,

$$\ln \mathcal{L}\left(\underline{\theta}\,;\mathcal{D}\right) = \sum_{i=1}^{N} \ln p_Y(y_i\,;\,\underline{\theta}) \tag{3.112}$$

We must express $p_Y(y;\underline{\theta})$ in terms of $p_Z(k)$ and $p(y|Z=k)$ in order to make explicit its dependence on the parameters.

$$p_Y(y\,;\,\underline{\theta}) = \sum_{k=1}^{K} p_{ZY}(y,k) \tag{3.113}$$

$$= \sum_{k=1}^{K} p_Z(k) p(y \mid Z{=}k) \tag{3.114}$$

$$= \sum_{k=1}^{K} \pi_k \,\mathcal{N}_k(y) \tag{3.115}$$

Then,

$$\ln \mathcal{L}\left(\underline{\theta}\,;\mathcal{D}\right) = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \,\mathcal{N}_k(y_i) \right) \tag{3.116}$$

The maximum likelihood optimization problem is then,

$$\begin{aligned} \underset{\{(\pi_k,\mu_k,\sigma_k^2)\}_K}{\text{maximize}} \quad & \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \,\mathcal{N}_k(y_i) \right) \\ \text{subject to} \quad & \sum_{k=1}^{K} \pi_k = 1 \\ & \pi_k \geq 0 \qquad k \in \{1 \ldots K\} \\ & \sigma_k^2 > 0 \qquad k \in \{1 \ldots K\} \end{aligned} \tag{3.117}$$

This is a very complicated optimization problem! Although the objective function is differentiable everywhere in the feasible set, it is not convex (or rather it is not concave since this is a maximization problem). Furthermore, the presence of an equality constraint implies that the feasible set has no interior, and hence it is unlikely that we will find a feasible stationary point.

A common trick for eliminating an equality constraint is to append it to the objective function using a so-called Lagrange multiplier λ. The theory of Lagrange multipliers is

beyond the scope of this course. We will simply accept that Eq. 3.117 is equivalent to:

$$\begin{aligned} \underset{\lambda,\underline{\theta}}{\text{maximize}} \quad & \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(y_i) \right) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \\ \text{subject to} \quad & \pi_k \geq 0 \qquad k \in \{1 \dots K\} \\ & \sigma_k^2 > 0 \qquad k \in \{1 \dots K\} \end{aligned} \tag{3.118}$$

The first order optimality conditions imply that the solutions to this problem are amongst the stationary points of the objective function, as well as the non-interior feasible points (i.e. points with $\pi_k = 0$ for some k). Next we find the stationary points by taking derivatives of the objective function with respect to each of the μ_k's, σ_k^2's, π_k's, and λ. For simplicity of notation, we denote the objective function with J:

$$J(\lambda, \underline{\theta}) = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(y_i) \right) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{3.119}$$

Derivative with respect to μ_r

Take the derivative of Eq. 3.119 with respect to μ_r, where r is any number in $\{1 \dots K\}$

$$\frac{\partial J}{\partial \mu_r} = \sum_{i=1}^{N} \frac{\pi_r}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(y_i)} \frac{\partial \mathcal{N}_r(y_i)}{\partial \mu_r} \tag{3.120}$$

$$= \sum_{i=1}^{N} \underbrace{\frac{\pi_r \, \mathcal{N}_r(y_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(y_i)}}_{\gamma_{ir}} \frac{(y_i - \mu_r)}{\sigma_r^2} \tag{3.121}$$

Here we have introduced the symbol γ_{ir} for the "responsibility" of class r for data point y_i. This quantity can be shown (via Bayes' rule) to be probability that data point i is of class r.

$$\gamma_{ir} = \frac{\pi_r \, \mathcal{N}_r(y_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}_k(y_i)} \tag{3.122}$$

Equating Eq. 3.121 to zero, we find a condition for stationarity.

$$\sum_{i=1}^{N} \gamma_{ir} \frac{(y_i - \mu_r)}{\sigma_r^2} = 0 \tag{3.123}$$

With $\sigma_r^2 \neq 0$, this leads to an expression for the optimal placement of the means of the components.

$$\mu_r = \frac{1}{N_r} \sum_{i=1}^{N} \gamma_{ir} \, y_i \tag{3.124}$$

Here we have defined N_r as the total responsibility of the r'th component.

$$N_r = \sum_{i=1}^{N} \gamma_{ir} \tag{3.125}$$

In words, the r'th component is centered at the weighted sample mean of the data points, with the weights set to the responsibilities of that component. It is easily shown that $\sum_{k=1}^{K} N_k = N$.

Derivative with respect to σ_r^2

Next we take a derivative of Eq. 3.119 with respect to σ_r^2.

$$\frac{\partial J}{\partial \sigma_r^2} = \sum_{i=1}^{N} \frac{\pi_r}{\sum_{j=1}^{K} \pi_j \mathcal{N}_j(y_i)} \frac{\partial \mathcal{N}_r(y_i)}{\partial \sigma_r^2} \tag{3.126}$$

Equating this to zero and skipping some of the details (which are easy, though tedious), we eventually get to a second stationarity condition (I encourage you to derive this):

$$\sum_{i=1}^{N} \gamma_{ir} \left(\frac{(y_i - \mu_r)^2}{\sigma_r^2} - 1 \right) = 0 \tag{3.127}$$

Which leads to

$$\sigma_r^2 = \frac{1}{N_r} \sum_{i=1}^{N} \gamma_{ri} (y_i - \mu_r)^2 \tag{3.128}$$

In words, the variance of the r'th component is the weighted sample variance of the data points, with weights set the responsibilities of that component.

Derivative with respect to π_r

Finally, we take a derivative of Eq. 3.119 with respect to π_r and equate it to zero.

$$\frac{\partial J}{\partial \pi_r} = \sum_{i=1}^{N} \frac{\mathcal{N}_r(y_i)}{\sum_{j=1}^{K} \pi_j \mathcal{N}_j(y_i)} + \lambda = 0 \tag{3.129}$$

Multiplying both sides by π_r:

$$\sum_{i=1}^{N} \gamma_{ir} + \pi_r \lambda = 0 \tag{3.130}$$

Using the definition of N_r, this implies that $\pi_r = -N_r/\lambda$. Separately, from the requirement that $\sum_k \pi_k = 1$, we find that $\lambda = -N$, and therefore,

$$\pi_r = \frac{N_r}{N} \tag{3.131}$$

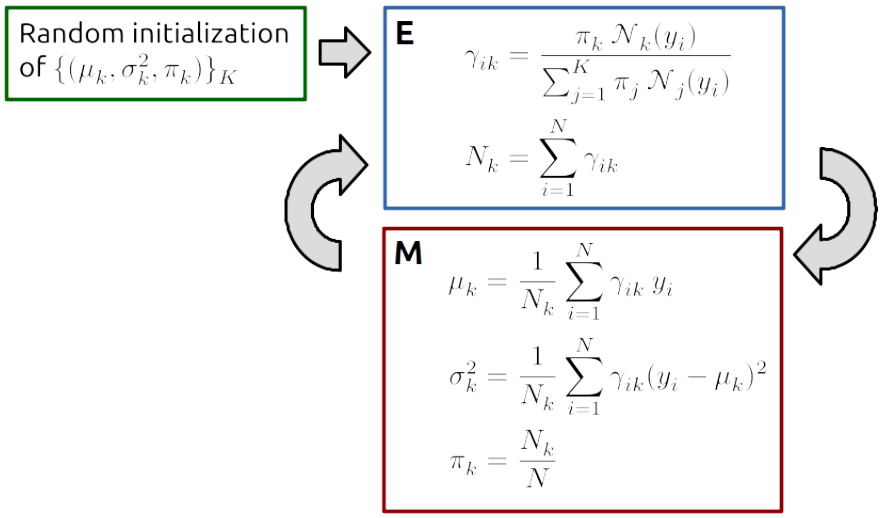

Figure 3.12: Expectation Maximization for Gaussian Mixture Models

In words, the marginal probability of class r is the ratio of the total responsibility of component r (N_r) to the number of points in the dataset (N). This justifies the interpretation of N_r as the "effective" number of data points in the r'th component.

To summarize, we have derived five formulas (Eqs. 3.122, 3.124, 3.125, 3.128, and 3.131) that characterize the local solutions of the maximum likelihood estimate for a Gaussian mixture. These formulas can be summarized as follows:

1. Given the marginal probabilities π_k's, compute the responsibilities γ_{ir} for each data point i and component r, using Eq. 3.122.

2. Compute the centroids μ_r and total responsibilities N_r for each component r, using Eqs. 3.124 and 3.125.

3. Compute the variances σ_r^2 and marginal probabilities π_r for each component r, using Eqs. 3.128 and 3.131

Taken together, these form a system of nonlinear equations that is difficult to solve. However notice that the first step assumes that the π_k's are given, and these are computed in the third step. This suggests that possible procedure for solving this system of equations might be to cycle through the three steps until the values converge. This turns out to be an excellent numerical algorithm for this problem, and it goes by the name of the *expectation-maximization or EM algorithm*. The details of the algorithm are shown in Figure 3.12.

Example 3.4.1. See `demo_gmm.py`.

3.5 Clustering algorithms and K-means

We saw in the previous section that fitting a Gaussian mixture to a dataset entails finding the mean and variance for each of the Gaussian components, and also the "responsibility" values

γ_{ik}. These responsibilities can be interpreted as membership values: the i'th data point has membership level γ_{ik} in the k'th Gaussian component. We can use the memberships to create *clusters* by assigning each data point to the component for which its membership is highest. Thus, the GMM procedure can be used to cluster the data, in addition to generating a model.

But what if we are only interested in the clusters, and not in the model? In this case the GMM approach may be overkill, and we may prefer an algorithm that focuses exclusively on the clustering task. There are many algorithms for doing this. Generally they go under the heading of "unsupervised learning", as opposed to "supervised learning", which we will cover starting in Chapter 4 and throughout the rest of the course. Whereas supervised learning algorithms teach machines to predict approximately correct answers by presenting them with examples of correct answers, unsupervised learning algorithms lack a concept of a "correct" answer. They are used simply to find patterns in the data. In the case of clustering algorithms, they find groups (a.k.a. clusters) of similar data points.

Gaussian mixtures were presented in the previous section in the single-input context ($D = 1$). Here we will generalize the presentation to multiple inputs ($D \geq 1$). The component means μ_k are now vectors in $\mathbb{R}^D$, and the variances σ_k^2 are $D \times D$ positive definite matrices which we will denote by Σ_k.

The K-means algorithm can be obtained via a simplification of GMM. The first step is to impose a simplified structure on the covariance matrices: Σ_k is required to be a diagonal matrix with equal entries ($\epsilon > 0$) along the diagonal.

$$\Sigma_k = \begin{bmatrix} \epsilon & 0 & \dots & 0 \\ 0 & \epsilon & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \epsilon \end{bmatrix} \tag{3.132}$$

Each of the component Gaussian pdfs are now radially symmetric; their level sets are circles. Each membership value γ_{ik} now depends only on the Euclidean distance from the data point to the mean μ_k of the Gaussian component. The next step is to take the limit as $\epsilon \to 0$. As we do this, the γ_{ik}'s migrate to their extreme values of 0 and 1. That is, the γ_{ik}'s become a hard indicator of cluster membership. The centroids μ_k are the mean of the members their cluster, and N_k is the integer number of points in cluster k.

Example 3.5.1. See `demo_kmeans.py`.

Let's now see what happens to the EM algorithm of Figure 3.12 when we apply the simplification just described. This is illustrated in Figure 3.13. As with GMM, the algorithm begins with a specification of the number of clusters K. The algorithm is initialized with a random placement of the centroids $\{\mu_k\}_K$. In the 'E' step, instead of computing responsibilities with Eq. 3.122, we now assign each data point to its nearest centroid. Then we find N_k as the number of data points assigned to the k'th cluster. The 'M' step then relocates the μ_k's to the mean of the points in each cluster. This is repeated until the clusters stop changing.

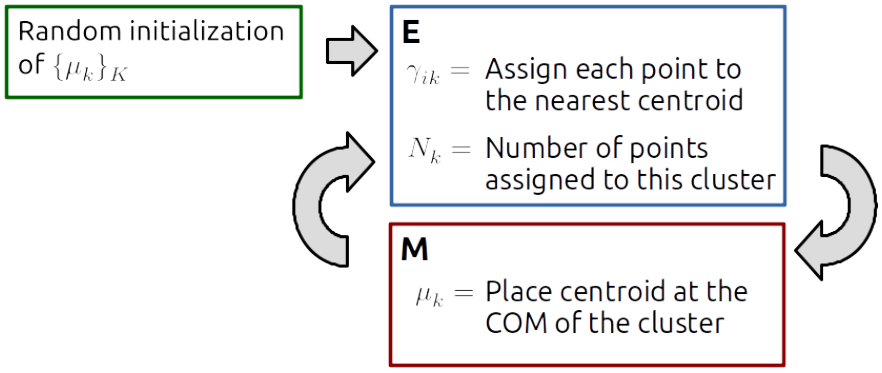

Figure 3.13: Basic K-means algorithm

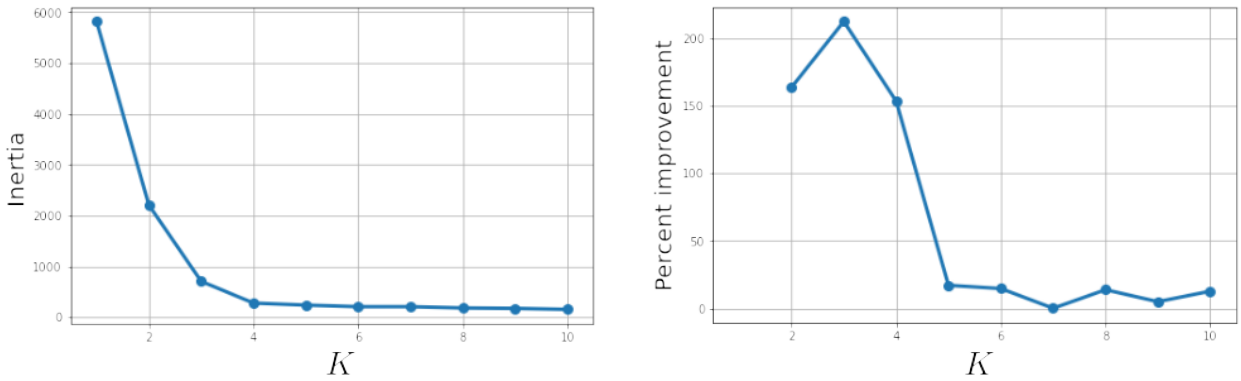

Figure 3.14: Sweep over K

Once convergence is reached, the result can be assured to be a local optimum, but not necessarily a globally optimal solution. It may be possible to reach a better solution by starting from a different initial placement of the centroids. It is very common, in fact, to do an "ensemble run", in which many executions of the basic algorithm are carried out, and on the best result (best local optimum) is kept.

Finally, we can run ensemble runs for different numbers of clusters. Figure 3.14 shows the result of running an ensemble of basic K-means models for values of K ranging from 1 to 10. It is clear that the best possible solution with $K = 5$ will be better than the best possible solution with $K = 4$, since $K = 4$ is achievable by leaving empty one of the clusters in $K = 5$. The optimal cost for K-means should therefore be a strictly decreasing function of K, which reaches zero in the extreme case of $K = N$ (the number of data points). To select a "best K", it is therefore necessary to judge the marginal benefit of increasing K by one, say from 4 to 5. The plot on the right shows these marginal gains as a function of K. For example, the cost is decreased by over 150% when going from $K = 1$ to $K = 2$ clusters. A further improvement of over 200% is attained when going to 3 clusters. However the improvement decreases to about 15% when going from $K = 4$ to $K = 5$. This means that $K = 4$ is likely a reasonable place to stop.

Chapter 4

Supervised learning: Static models with inputs

In Section 3.1.6 we described the maximum likelihood approach to building models of systems with no inputs. The steps were as follows.

1. Collect an iid dataset.
2. Propose a parametric family of distributions.
3. Find values of the parameters that maximize the likelihood of the model given the data (Eq. 3.55).

We now extend the setting to include systems with *inputs*. This is illustrated in Figure 4.1. The input x is a vector in $\mathbb{R}^D$, with values corresponding to each of the D inputs of the system. We will study two important problems in this context. The *inference problem* (left side of Figure 4.1) is analogous to the maximum likelihood problem of section 3.1.6, except that now, instead of seeking a single distribution, we seek distributions of the output *for every value of the input*. In other words, we wish to approximate the conditional distribution $Y|X = x$, for every value of x. Notice that we have assumed the input x to be a *known* vector of values. This cannot strictly be true. The input, just like any other measurement, is subjected to measurement error. This input uncertainty may simply be ignored, or it may be absorbed into the box and modeled along with the rest of the system.

As in the previous chapter, estimating a *distribution* for the output enables some powerful analyses. For example, we can compute confidence intervals and conduct hypothesis tests

Figure 4.1: Closed box models with inputs

on the parameters of the model and on its predictions. We will see how to do this in the context of linear regression in Chapter **??**.

Sometimes, however, we are only interested in predicting the output of the system given its input, and not in the deeper analyses. We can certainly make such "point" predictions if we have already solved the inference problem, for example by taking the mean of the estimated conditional distribution. But if that is all we want, then it may be preferable to forego the inference problem and focus on building a *prediction function* $h(x)$; i.e. one that simply returns an output value for each input. This is depicted on the right hand side of Figure 4.1[1]. Most of the rest of the course will be concerned with techniques for solving the prediction problem. We begin in this chapter with an overview of a general framework called *supervised learning*, which can be applied to both the inference and the prediction problems.

The term "supervised learning" refers to algorithms for building inference or prediction models based on samples of the system's inputs and output. The approach follows the three steps listed earlier for building probabilistic models using maximum likelihood. For the prediction problem the steps are:

1. Collect an iid dataset.
2. Propose a parametric family of *input/output functions*.
3. Find the values of the parameters that *minimize the loss* of the model given the data.

Notice that the "distributions" from the maximum likelihood problem have been replaced with "input/output functions", and instead of maximizing the likelihood, we are minimizing a loss function. There are many candidate model families to choose from in step 2. These include K-bins (Chapter XXX), K-nearest neighbors (Chapter XXX), linear regression (Chapter XXX), logistic regression (Chapter XXX), decision trees (Chapter XXX), support vector machines (Chapter XXX), neural networks (Chapter XXX), and others. In this chapter we cover high level notions that apply generally to the framework, irrespective of the chosen model family.

4.1 The data

The first step is to collect a dataset $\mathcal{D}$ consisting of N samples of inputs and the corresponding output of the system. This is expressed in probabilistic terms as an iid sample from an underlying joint distribution (X, Y):

$$\mathcal{D} = \{(x_i, y_i)\}_N \overset{\text{iid}}{\sim} (X, Y) \tag{4.1}$$

X is the multivariate collection of the D random inputs: $X = (X^1, \ldots, X^D)$. Notice that we are using superscripts for the inputs, and subscripts for the input-output samples. Hence

[1] A third possibility is to build a model of the joint distribution (X, Y). This is called a *generative* mode, and it is beyond our scope.

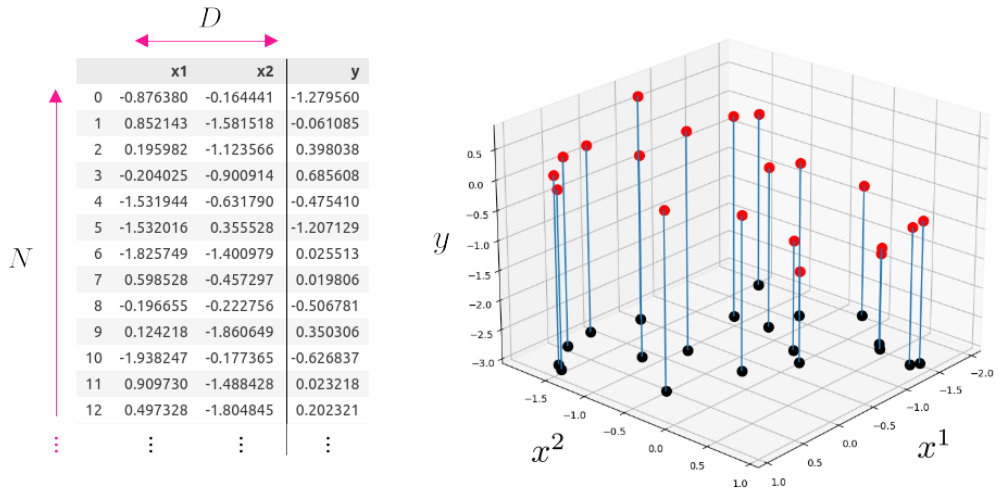

Figure 4.2: Two views of a dataset with $D = 2$.

the i'th sample can be written as

$$(x_i^1, \ldots, x_i^D, y_i) \tag{4.2}$$

These data can be organized into a table with N rows and $D + 1$ columns, as shown in [2]Figure 4.2. Each row corresponds to a sample. The first D columns hold the inputs, and the right-most column holds the output. The fact that the samples are iid means that we are free to shuffle the rows. The right side of Figure 4.2 shows a scatter plot of the data. Each data point is thought of as being generated in a two-step process, captured by the decomposition of the joint input-output distribution into a marginal times a conditional:

$$p_{XY}(x, y) = p_{Y|X=x}(y)\, p_X(x) \tag{4.3}$$

First, an input sample $x_i = (x_i^1, \ldots, x_i^D)$ is drawn from p_X. This corresponds in the figure to sampling a black dot in the horizontal plane. This sample determines a conditional distribution along the vertical axis, which generates y_i (the corresponding red dot).

Generative models are ones that capture both parts of this data generating process: input generation and its transformation through the system. *Discriminative* models only capture $p_{Y|X=x}$. They take the input as given, and are not concerned with how it was generated. This is obviously a simpler problem to solve, and it is the one that we will focus on in this course. Confusingly, in the classification setting both of these model types are referred to as "generative", and the term "discriminative" is reserved for models that return only a selected class and not a distribution over classes.

Prediction problems can be split broadly into two types: *classification* problems and *regression* problems. In a classification problem, the sample space of the output is *categorical*, meaning that the model must produce a category or a "label" for each input. For example, the classification of vehicles into bicycles, scooters, and mopeds according to their weight. In

[2]demo_2D.py

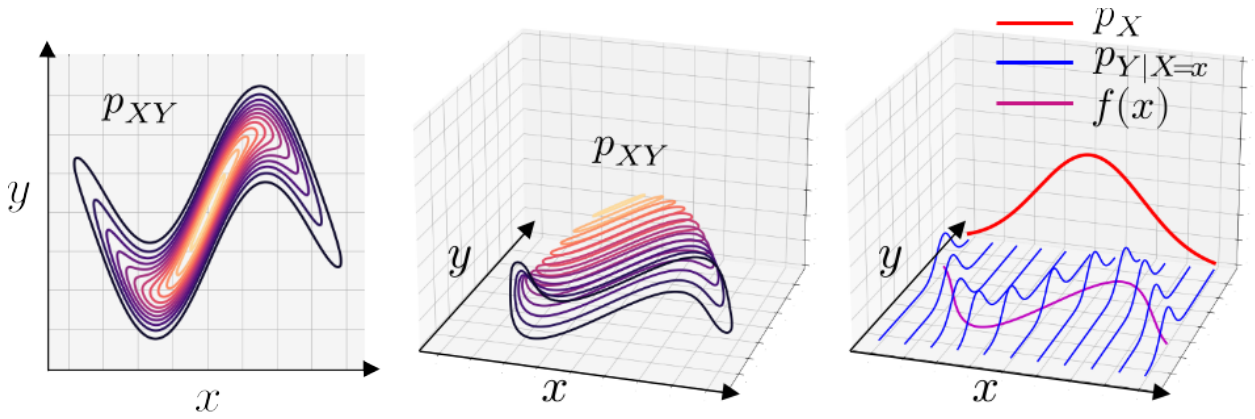

Figure 4.3: 1D regression problem.

regression problems, the output is *numerical*, typically a real number. The table of Figure 4.2 corresponds to a regression problem. Two important distinctions between labels and numbers are a) labels are discrete and b) labels lack order. Systems that produce *integer* numbers are an intermediate case: they are discrete (like labels) but ordered (like real numbers). There are specialized methodologies for such "ordinal" problems, but they are beyond the scope of this course. Here we focus on the two basic cases: real-valued regression and label-valued classification.

Let's consider the regression problem for a system with a single input ($D = 1$). Figure 4.3 shows views of the joint distribution in the leftmost and middle plots, and its decomposition into the product of p_X (in red) and $p_{Y|X=x}$ (in blue) in the rightmost plot. Consider the discriminative case, where our goal is only to approximate the conditional distribution $p_{Y|X=x}$. The input-to-output transformation represented by $p_{Y|X=x}$ can itself be understood (without loss of generality) as a two-step process. First the input is transformed by a deterministic function $f(x)$ into an expected output, and then the expected output is corrupted by zero-mean noise. This is expressed as follows,

$$y = f(x) + \varepsilon \tag{4.4}$$

where $f(x)$ is the expected value of the output for input x,

$$f(x) = E[\, Y \mid X = x \,] \tag{4.5}$$

and ε is zero-mean noise ($E[\varepsilon] = 0$). The common interpretation of Eq. 4.4 is that $f(x)$ represents the "true" model of the system, while ϵ captures uncertainties such as measurement errors and other unknown quantities. In the rightmost plot of Figure 4.3, $f(x)$ is the purple line, and the distributions of ε for each x are shown in blue. In the most general setting, the shape of the pdf of ε may depend on x. That is, each of the blue lines in the figure may have a different shape (not just a different location). However it is customary to assume that this is not the case, and that ε is independent of the input.

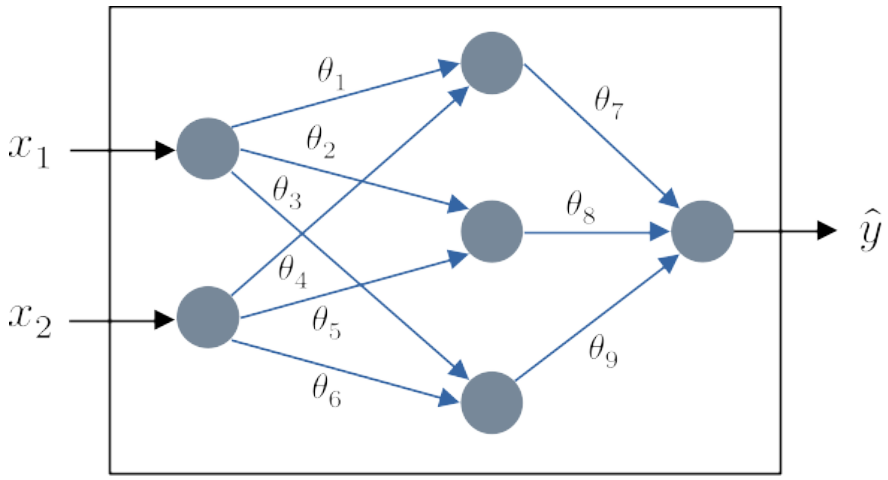

Figure 4.4: A tiny neural network with 6 vertices and 9 weighted edges.

The prediction problem is then to construct a function $h(x)$ that approximates $f(x)$. The inference problem requires both the estimation of $f(x)$ and of the distribution of ε.

4.2 Parametric families of models

We focus on the prediction problem; i.e. to construct a prediction function $h(x)$ that closely matches the unknown $f(x)$. We use $\mathcal{H}$ to denote a parameterized family of prediction functions. P is the number of parameters that parameterize $\mathcal{H}$, and $h(x;\theta) \in \mathcal{H}$ is a particular member of $\mathcal{H}$ with the parameters set to $\theta \in \mathbb{R}^P$. (We are dropping the underline from the $\underline{\theta}$ of the previous chapter. θ is always vector-valued from now on). For example, $\mathcal{H}$ might be the family of parabolas, i.e. all functions that map $x \in \mathbb{R}$ to $\theta_0 + \theta_1 x + \theta_2 x^2$, where θ_0, θ_1, and θ_2 are real numbers. This family has $P = 3$, because it is parameterized by $(\theta_0, \theta_1, \theta_2) \in \mathbb{R}^3$.

Example 4.2.1. Figure 4.4 shows a tiny neural network. The details of how it works are not important for now and are covered in Chapter **??**. This neural network has 9 tunable parameters $\theta_1 \ldots \theta_9$, corresponding to the "weights" on each of its edges. The family of neural network with this particular architecture has $P = 9$ and is parametrized by $\theta \in \mathbb{R}^9$.

Each parameter vector θ yields a prediction function $h(x;\theta)$. We denote with $\hat{y}$ the prediction for input x:

$$\hat{y} = h(x;\theta) \tag{4.6}$$

The semicolon in the notation separates the input of the function (x) from its parameters (θ). Step 3 of the procedure is to find the "best" prediction function h from the proposed family of prediction functions $\mathcal{H}$. The performance of a particular prediction function is gauged in terms of the expected prediction error. That is, in terms of the expected "distance" between the predicted value $\hat{y}$ and the true value y, for each input value x drawn from the distribution of inputs X. This distance is measured using a *loss function*.

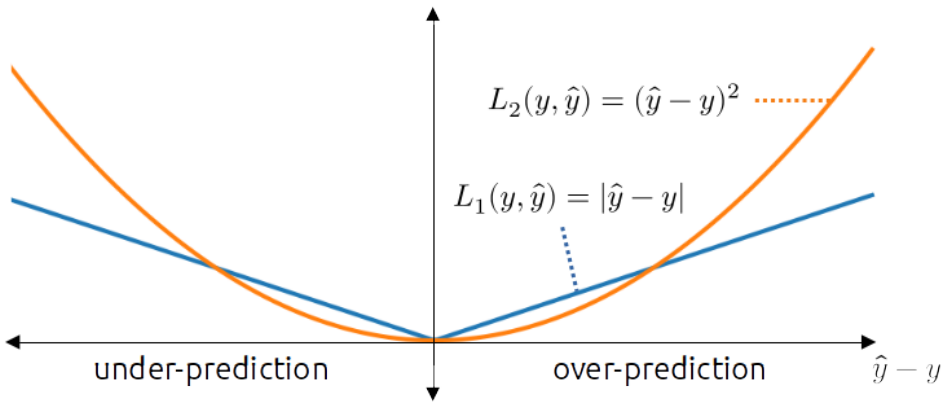

Figure 4.5: Example loss functions.

4.3 Loss function

The *loss function* measures the difference between the predicted output $\hat{y}$ and the actual output y, for a given input x. A valid loss function has these characteristics:

1. It is never negative.

2. It is zero (or small) whenever the predicted output $\hat{y}$ matches the true output y.

3. It grows with the size of the prediction error.

Figure 4.5 shows two examples of loss functions used in regression problems: the *squared loss* (a.k.a. L_2 loss), and the *absolute value* loss (a.k.a. L_1 loss). These are only used in regression problems and are *not* appropriate for classification problems. Loss functions used in classification, such as the cross-entropy loss, are introduced in Chapter **??**.

Our choice of L_1 or L_2 or some other loss function reflects our preferences about the resulting prediction errors. For example, both L_1 and L_2 are symmetric with respect to the vertical axis. This means they penalize over-prediction and under-prediction equally. If for a particular problem we deem over-prediction as worse than under-prediction, then we can use a lopsided loss function that penalizes positive errors more harshly than negative errors.

The loss function can also be used to control the occurrence of large outlier errors. Models trained with the L_2 loss will tend to produce errors that are more "nicely behaved" (i.e. normally distributed, with few large positive or negative "spikes") than those resulting from the L_1 loss function. This is because the L_2 penalty rises more quickly and therefore penalizes outliers more severely than L_1. On the other hand, models trained with the L_1 loss will tend to produce a lower average error, relative to the L_2 loss. This is because the total L_1 loss is proportional to the average absolute error. Despite this advantage, L_2 is usually preferred due to numerical and analytical considerations.

4.4 Optimization problem

We can now state the general supervised learning approach to the prediction problem as an optimization. The dataset used here is called the "training" dataset $\mathcal{D}_{\text{train}}$.

Given $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_N \overset{\text{iid}}{\sim} (X, Y)$,

$$\begin{aligned} \underset{h \in \mathcal{H}}{\text{minimize}} \quad & \sum_{i=1}^{N} L(y_i, \hat{y}_i) \\ \text{subject to} \quad & \hat{y}_i = h(x_i) \quad i = 1 \ldots N \end{aligned} \tag{4.7}$$

or equivalently, in terms of θ,

$$\begin{aligned} \underset{\theta \in \mathbb{R}^P}{\text{minimize}} \quad & \sum_{i=1}^{N} L(y_i, \hat{y}_i) \\ \text{subject to} \quad & \hat{y}_i = h(x_i; \theta) \quad i = 1 \ldots N \end{aligned} \tag{4.8}$$

or more briefly,

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^P}{\text{argmin}} \quad \sum_{i=1}^{N} L(y_i, h(x_i; \theta)) \tag{4.9}$$

We have seen a similar optimization problem already in Section 2.4.1 when we introduced stochastic gradient descent. Indeed, SGD is the most widely used (although not the only) numerical algorithm for solving supervised learning problems. However SGD can only be used if the gradients can be computed. Let's consider this issue more closely.

Recall that the update equation for SGD (Eq. **??**) makes use of ∇L_θ; the gradient of L with respect to the parameters. Using the chain rule, we find that this can be expressed as a product of two terms:

$$\nabla_\theta L = \left(\frac{\partial L}{\partial \hat{y}} \right) \nabla_\theta \, h \tag{4.10}$$

The first term is the partial derivative of the loss function with respect to its second argument. This is a scalar quantity, and it corresponds simply to the slope of the curve in Figure 4.5. We can see that for the L_2 loss, the slope equals the "residual", $\hat{y} - y$, while for L_1 it is -1 for negative errors and $+1$ for positive errors:

$$\frac{\partial L}{\partial \hat{y}} = \begin{cases} \hat{y} - y & \text{for } L_2 \text{ loss} \\ \text{sign}(\hat{y} - y) & \text{for } L_1 \text{ loss} \end{cases} \tag{4.11}$$

This discontinuity in the gradient of the L_1 loss precludes the use of SGD. We will not consider it further in the course for this reason. The second term in Eq. 4.10 ($\nabla_\theta h \in \mathbb{R}^P$) is the vector gradient of the model with respect to its parameters. The difficulty of computing this term depends on the model family that we have selected. For example, finding the

gradient for the family of parabolas $h(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2$ is easy.

$$\nabla_\theta h = \left(\frac{\partial f}{\partial \theta_0}, \frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}\right) \tag{4.12}$$

$$= (1, x, x^2) \tag{4.13}$$

However computing the gradient of a neural network is more difficult, and was in fact a major obstacle to their use until an efficient implementation of the *backpropagation* algorithm became available in the 1980's. More on this in Chapter **??**.

4.5 Assessing model performance

Once we have solved the optimization problem and obtained a prediction function $h(x;\hat{\theta})$, the next question is how to assess the quality of the result. We might think that by virtue of the optimization problem, $h(x;\hat{\theta})$ is the best amongst its family. We will see that this is not necessarily so, due to the phenomenon of *overfitting*, and we will introduce techniques for detecting and avoiding this problem. We begin with basic metrics that are insensitive to overfitting: MSE, RMSE, MAE, and R^2. These pertain to regression problems; metrics for classification problems are introduced in Chapter **??**.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad \text{... mean squared error} \tag{4.14}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \qquad \text{... root mean squared error} \tag{4.15}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \qquad \text{... mean absolute error} \tag{4.16}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \qquad \text{... mean absolute percentage error} \tag{4.17}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \qquad \text{... coefficient of determination} \tag{4.18}$$

First, on the MSE, the definition of equation Eq. 4.14 should be distinguished from the probabilistic quantity defined in Eq. 3.36. The probabilistic MSE applies to estimators in general. The prediction function $h(x,\hat{\theta})$ can be regarded as an estimator of the true output $f(x)$. The randomness arises from the sampling of the training data. This uncertainty affects $\hat{\theta}$ through the training process. The evaluation of this uncertain prediction function on a *fixed* input x produces variations in $\hat{y}$. Denoting with $\hat{Y}(x)$ the corresponding random

variable (conditioned on each fixed x), we can write its bias and variance:

$$Bias[\hat{Y}(x)] = E[\hat{Y}(x)] - f(x) \tag{4.19}$$

$$Var[\hat{Y}(x)] = E[(\hat{Y}(x) - E[\hat{Y}(x)])^2] \tag{4.20}$$

Its MSE is also a function of x:

$$\text{MSE}(x) = E[(E[\hat{Y}(x)] - f(x))^2] \tag{4.21}$$

If we knew the distribution p_X, we could calculate an *aggregate* MSE, by averaging MSE(x) over all possible inputs. Call this $\overline{\text{MSE}}$. The numerical quantity of Eq. 4.14 is an average over the input space with the prediction function held fixed. This is an unbiased estimate of $\overline{\text{MSE}}$. Averages of the MSE of Eq. 4.14 converge to $\overline{\text{MSE}}$ as we generate new models from freshly drawn training data.

As a performance metric, the MSE can be difficult to interpret because its units are the square of the units of y. For reporting purposes it is more common to use the RMSE (Eq. 4.15), which is simply the square root of the MSE. Both the MSE and the RMSE are increasing functions of the total L_2 loss. Hence $h(x, \hat{\theta})$ will be the minimizer of both MSE and RMSE when L_2 is used in the optimization problem. The MAE (Eq. 4.17) computes the average of the absolute values of the residuals. Its units are the same as the output, which makes it easier to interpret than MSE. It is compatible with the L_1 loss function in the same way as MSE and RMSE are compatible with L_2. The MAPE goes a step further than MAE in its interpretability, since it is unitless. A MAPE of 0.2 means that the prediction is 20% from the true value on average.

The remaining difficulty of interpretation relates to the *scale* of the metric. Whether a MAPE of 0.2 is good or bad depends on the difficulty of the problem. The *coefficient of determination* R^2 addresses this issue by comparing the error incurred by the model with the error of a baseline model $\bar{h}(x) = \bar{y}$, where $\bar{y}$ is the average of the outputs in the training dataset: $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. An R^2 of 1 indicates a "perfect" model; one that perfectly hits all of the points in the training dataset. R^2 can never exceed 1. $R^2 = 0$ corresponds to the baseline model. Hence we expect the R^2 to fall between 0 and 1, with larger values being preferred. Negative R^2 indicates a model that performs *worse* than the baseline model.

These five metrics measure the residuals obtained during the training process, but they are not necessarily good estimates of *future* model performance. Figure 4.6 illustrates the problem. Here the blue dots are the data used to train the model ($\mathcal{D}_{\text{train}}$), and the red plus signs are samples that were withheld. We call this the *test* dataset $\mathcal{D}_{\text{test}}$. Each of the four plots shows the optimal prediction function selected from four different families: linear functions, cubics, order five polynomials, and order 10 polynomial. These families are nested in the sense that the cubics include all of the linear functions as a sub-family, the order five polynomials include the cubics, etc. Hence, the "training error" for the optimal cubic polynomial (i.e. the optimal $L2$ loss) will necessarily be better (or no worse) than that of the linear functions. Similarly, the optimal order five polynomial is better than the optimal cubic, and so on. However the same metric applied to the *test* data tells a different

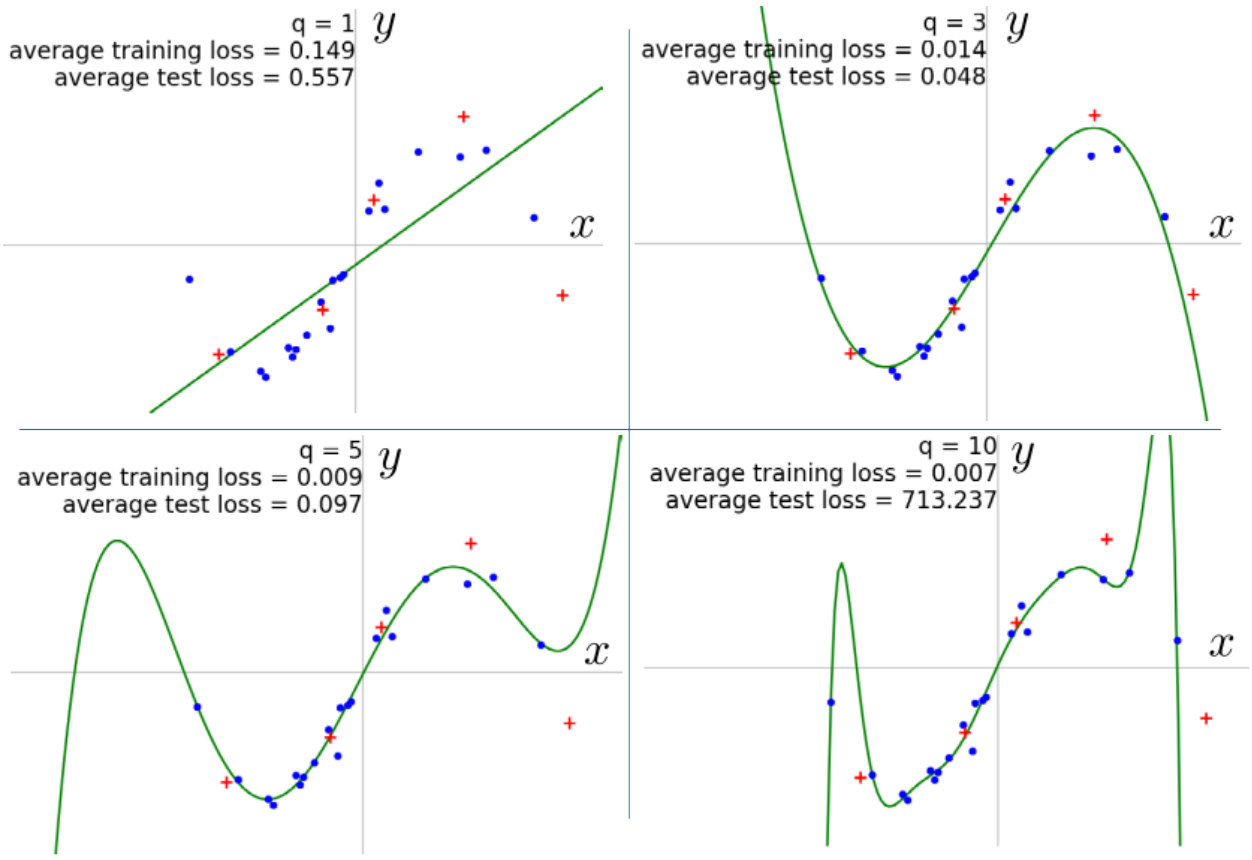

Figure 4.6: 1D regression problem. See `demo_poly.py`.

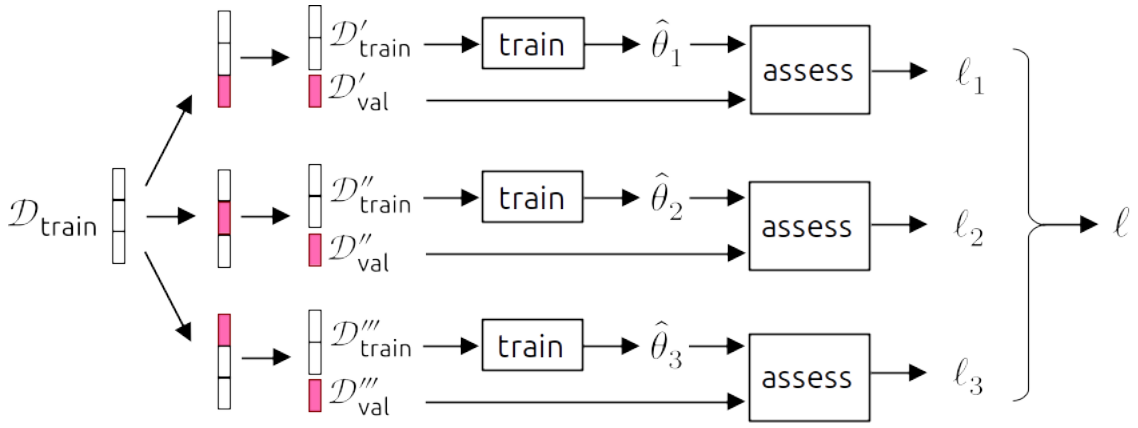

Figure 4.7: K-fold cross validation with $K = 3$.

story. The cubic polynomial is the best in terms of the "test error", while the order 10 polynomial, which was best on training, performs terribly. The order 10 model is severely *overfitted*. Overfitting occurs when the model family used in training is too flexible for the system being modeled. In the example, the underlying system function happens to be a sine wave, which is similar to a cubic function in the regime covered by the dataset. The added flexibility of tenth order polynomials allows them to more closely track the training data, but this turns out to be detrimental to the test error.

How can we detect whether a model is overfitted? The most comprehensive and universal approach is what we have just described: to withhold a portion $\mathcal{D}_{\text{test}}$ of the data from the training process, and to use it to estimate future model performance. This works well if we have sufficient data. Otherwise, if data is scarce and data collection expensive, then there are metrics that explicitly penalize model flexibility via some proxy, such as the order of the polynomial. Examples of these include Akaike's information criterion (AIC) and the Baysian information criterion (BIC). These approaches tend to be model-specific, and will not be covered here. A more universal and data-centered technique for mitigating data-scarcity is called *K-fold cross-validation*, and we cover this next.

4.5.1 K-fold cross-validation

In situations of data scarcity, it is reasonable to want to use all of the available data for training the model. But this leaves us with no data for testing, which raises the possibility of overfitting. K-fold cross-validation is a model assessment technique that produces an unbiased estimate of model performance while allowing all of the data to be used for training. Figure 4.7 illustrates the approach with K set to 3. The process begins by splitting $\mathcal{D}_{\text{train}}$ into K equal parts. $K-1$ of those parts constitute a training set $\mathcal{D}'_{\text{train}}$, and the remainder, $\mathcal{D}'_{\text{val}}$, is known as the *validation data*. The "train" block in the figure takes a training data as input, runs SGD or some other training algorithm, and returns an optimal set of parameters $\hat{\theta}_1$. This model is assessed by evaluating any of the aforementioned performance metrics using the validation data. This produces an *unbiased* estimate of model performance (ℓ_1),

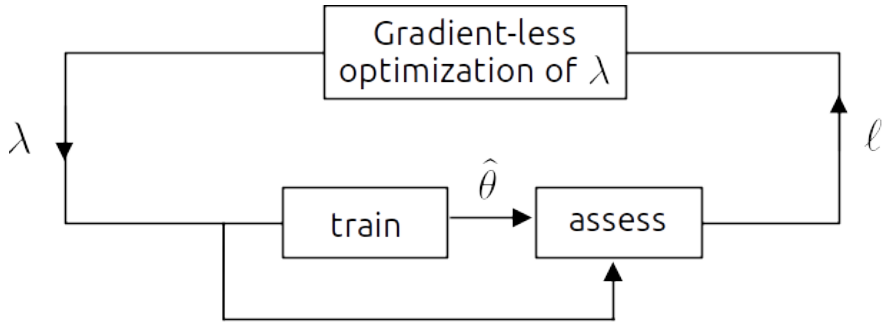

Figure 4.8: Hyper-parameter search.

since $\mathcal{D}'_{\text{val}}$ was not seen by the training step. The process is repeated K times, each with a different $1/K$'th portion of $\mathcal{D}_{\text{train}}$ as the validation dataset. Finally, the K performance assessments $\ell_1 \ldots \ell_K$ are averaged to obtain an overall estimate:

$$\ell = \frac{1}{K} \sum_{k=1}^{K} \ell_k \tag{4.22}$$

The question that remains is which if the K models (i.e. which of $\hat{\theta}_1$ through $\hat{\theta}_K$) should be kept? The answer is none. K-fold cross-validation is used only to estimate the performance of the model. The model parameters themselves are obtained by running the training algorithm on the full dataset $\mathcal{D}_{\text{train}}$.

4.6 Hyper-parameters

A parameter p of a model is a *hyper-parameters* when it cannot be optimized by stochastic gradient descent because the loss function cannot be differentiated with respect to p. The order of the polynomial in Figure 4.6 is an example of a hyper-parameter. We cannot take a derivative of L with respect to the polynomial order because a) it is an integer, and b) modifying the order changes the *formula* for h by adding new term. This captures the two main reasons why a parameter may be considered a hyper-parameter; either because it is not real-valued or because it is a "structural" parameter of the model family itself. In terms of notation, we will reserve θ for the tunable parameters, and use λ for the vector of hyper-parameters.

Figure 4.8 illustrates the general approach to optimizing hyper-parameters. The main idea is to use a *gradient-less* optimization method, such as the ones introduced in Section 2.5. The optimization method will advance by suggesting new λ's to evaluate. These are passed to the "train" block, which runs SGD to obtain the parameter estimate $\hat{\theta}$ corresponding to the given λ. This is then evaluated in the "assess" box using cross-validation, and the result (ℓ) is returned to the search algorithm. This continues until an optimal value of λ is found.

Bibliography

[1] https://math.stackexchange.com/questions/72975/variance-of-sample-variance. [Online; accessed September 2022].

[2] https://en.wikipedia.org/wiki/Chi-squared_distribution. [Online; accessed September 2022].

[3] https://en.wikipedia.org/wiki/Laplace_distribution. [Online; accessed September 2022].

[4] https://www.statskingdom.com/z_table.html. [Online; accessed September 2022].

[5] https://www.statskingdom.com/t_table.html. [Online; accessed September 2022].