

Paul Fieguth

# An Introduction to Pattern Recognition and Machine Learning

 Springer



## Overview

Pattern Recognition and Machine Learning were once something of a niche area, which has now exploded to become one of the hottest areas of study and research. Students from nearly every field of study clamour to study pattern recognition courses, researchers in nearly every discipline seek ways in which machine learning/deep learning might be applied to their domain, and companies seem enthusiastic in their rush to incorporate “intelligence” (of some sort) into fridges, toasters, and washing machines.

### What Is Pattern Recognition?

In many ways, pattern recognition is not at all a new discipline. It has *long* been important in statistics to perform statistical inference of some unknown quantity on the basis of related measurements, what is broadly referred to as an inverse problem. If the unknown quantity is *continuous* in nature — height, speed, time interval etc. — then the field of study regarding such inference is referred to estimation theory. In contrast, if the unknown quantity is *discrete* in nature — species of bird, type of protein, status of a machine (normal/broken) — then the inference is known as pattern recognition.

That is, pattern recognition is entirely focused on determining a particular identity (the pattern “class”) based on measured information, and the process which selects the class is known as a classifier:

Measurements  $\xrightarrow{\text{Classifier}}$  Inferred Class

The central task of pattern recognition consists of two parts:

FEATURE EXTRACTION: Which measurements are relevant to the classification process, and which measurements are irrelevant? Can we find *features*, some

function of the given measurements, that more compactly and reliably encode the salient information?

CLASSIFICATION: Given measurements or features, how do we determine their associated *class*, their underlying discrete identity?

A great many classification approaches have been developed, from the very simple (straight line or plane boundary between two classes), mathematically rigorous (statistically optimal classifiers), to much more advanced (large ensembles or aggregates).

Deep learning and associated deep neural networks dominate much of the related topics of pattern recognition, machine learning, artificial intelligence, and computer vision. Neural networks have been particularly successful at high-level language and vision problems, such as text translation, object recognition, or video captioning, but at a cost of very high computational complexity and, in most cases, absolutely no interpretability regarding what the network is doing or how it is accomplishing its task. This text will offer a systematic study of the mathematical, methodological, and conceptual developments that ultimately lead to deep networks, which are therefore discussed in their logical place as part of nonlinear ensembles of classifiers in [Sections 11.4](#) and [12.2](#), however there are a great many other books which discuss the practical aspects of programming network architectures and network learning.

### How to Read This Book

The chapters are organized conceptually and pedagogically to be read/taught from beginning to end, so that approach would be recommended for anyone who is willing to invest the time.

Furthermore, as is perhaps true of most subjects, pattern recognition is best learned by *doing*, and so a substantial number of sample problems are offered at the end of each chapter. Pattern recognition can be thought of as two parallel disciplines, one analytical/mathematical, and the other experimental/computational. This text tries to provide a balance of both, with an emphasis on conceptual understanding (with supporting derivations in [Appendix D](#)), but also with a fully worked numerical/computational lab study at the end of every chapter and programming/computational questions at the end of each chapter.

For those readers who are inclined to jump around in a text to get an appreciation for and an overview of pattern recognition, you might begin by looking at the examples and case studies (listed on [page xv](#)) since these are written in an encapsulated style, intended to be readable on their own.

For those readers eager to jump in and quickly actually *do* some pattern recognition, you should probably look through [Chapter 2](#) to understand the overall concepts and terminology, and then progress to distance-based classification in [Chapter 6](#), since distance-based approaches are fairly intuitive and quick to implement. The computational lab study at the end of every chapter may also help as a quick introduction to basic techniques.

Finally, for those readers who want to jump straight into Deep Neural Networks, there are many other textbooks and online guides that can do this for you. [Section 11.4](#) does discuss nonlinear ensembles, which includes deep networks, and which the reader could take a look at (along with that chapter's [Further Reading](#)), however the preceding 319 pages aim to give much deeper insight into the pattern recognition problem, and so the reader will be missing a great deal of context and understanding if jumping straight to large nonlinear networks.

Most of this text, and nearly all of the examples and case studies, are intended to be accessible and appreciated without necessarily following the details of the mathematics. To avoid cluttering the main body of the text, many of the mathematical details and derivations were moved to [Appendix D](#), however a deeper mathematical understanding is one of the central goals of this text, and the reader is very much encouraged to follow the derivations in [Appendix D](#) in parallel with the text. For those readers whose mathematics background needs refreshing, the appendices provide most of the material needed on algebra ([Appendix A](#)), random vectors ([Appendix B](#)), and optimization ([Appendix C](#)).

This book is, to be sure, only an introduction, and there is a great deal more to explore. Directions for further reading are proposed at the end of every chapter.



# Introduction to Pattern Recognition

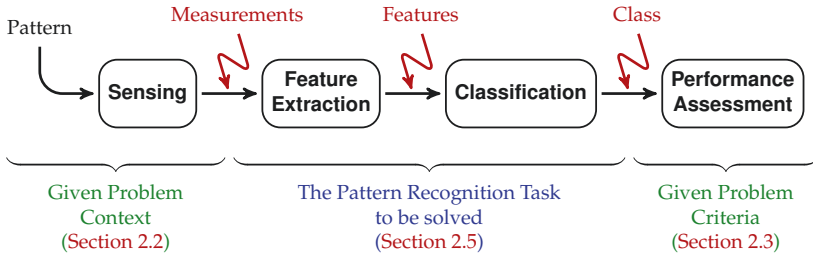
## 2.1 What Is Pattern Recognition?

*Pattern recognition is a process by which some input is measured, analyzed, and then classified as belonging to one of a set of classes.*

Although this opening definition may sound somewhat abstract, in actual fact the process of pattern recognition and classification is a continual, never-ending aspect of every-day human existence:

<b>Pattern recognition task</b>	<b>Possible classes</b>
What is in front of you as you walk?	Door vs. Window Sidewalk vs. Road
What music are you listening to?	Familiar or Unfamiliar Genre (Rock, Classical, ...) Name of Composer or Group
Is the traffic intersection safe to cross?	Green vs. Red light Pedestrian Walk vs. Stop Car Present vs. Not Present
Reading a page in a textbook	Letters of the Alphabet Text vs. Graphics Languages
You smell something in your apartment ...	Cookies finished baking? (Yes/No) Is something burning? (Yes/No) Wet dog/Skunk/Dead mouse/...

As a human experience, pattern recognition refers to a perceptual process in which some form of sensory input is sensed, analyzed, and recognized



**Fig. 2.1.** PATTERN RECOGNITION: A pattern is sensed, giving rise to measurements, from which feature are extracted and given to a classifier, whose job it is to select the class associated with the sensed pattern. In most cases the measurements and the classes are characterized (green) by the given problem definition, and the pattern recognition task (blue) is to infer a strategy for feature extraction and classification.

(classified), either subconsciously (by instinct) or consciously (based on previous experience). Patterns may be presented in any sensory modality: vision, hearing, touch, taste, or smell.

As a technical discipline, pattern recognition refers to a process in which an input object is measured, analyzed, and classified by a machine as being more or less similar to some class in a set of classes.

There are nearly endless contexts to which pattern recognition may be applied, from broad problems such as object recognition, to human-mimicking tasks such as text recognition (as in [Example 2.1](#)) or face recognition, to applications such as medical diagnosis or fault detection in physical systems. Motivating contexts include cases in which the amount of data may be too large, the number of decisions too great, or the pattern distinctions too imperceptible for a human to effectively undertake the classification.


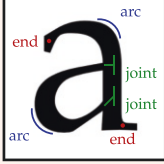
The conceptual framework for pattern recognition, whether human-based or machine-based, is illustrated in [Figure 2.1](#). In general,

- for a given definition of the information at hand (the measurements, [Section 2.2](#)),
- and some definition of what problem we wish to solve (the classes, [Section 2.3](#)),
- the goal of pattern recognition is to provide a machine with a kind of perceptual capability to automatically extract useful information from measured data (classification, [Section 2.4](#)).

The following three sections address these issues, in turn.

### Example 2.1: Pattern Recognition of Text

When you read a printed character on this page, your eye images the character on your retina, where it is sensed and converted into a neural representation which is subsequently analyzed in your brain. If your memory indicates prior experience with the character, or with a symbol sufficiently similar to it, you perceive or recognize the symbol, associating a label and meaning with it. What appears to us to be a relatively simple process becomes ever more impressive as we attempt to design a machine which might accomplish the same task, known as optical character recognition (OCR). What features might a machine use in recognizing text?

Pattern	Attributes to Measure	Measurements	Strengths and Weaknesses
"a" →		Vector of pixel values	Fast, easy, explicit Sensitive to changes in font style, size, and rotation
"a" →		Vector of shape properties	Robust to changes in size and rotation Complicated features to extract
"a" →	Complex Nonlinear Algorithm	Vector of values, but with no intuition	Possibly very flexible May be very hard to learn Difficult to analyze

For a machine needing to deal with printed characters in only a single font, a simple classification scheme might sense an array of pixel intensity values (top example, above) and compare it to arrays stored in memory corresponding to known characters. Such a template matching approach is simple and fast, but will be subject to a variety of limitations: any distortion, translation, rotation, scale change, or even lighting variations will affect the degree to which the given measurement matches a template.

If the machine must accept a variety of fonts, as humans would in reading, a memorized template matching scheme becomes impractical, since an unreasonable number of templates would need to be stored and compared. An alternate approach is to seek distinctive character features (middle example, above), such as strokes, arcs, loops etc. It might also be possible to train a black-box strategy (bottom), whereby some complex nonlinear function is learned, producing features which no longer have any intuitive meaning regarding recognizing characters, but which turn out to be effective features in classifying typed characters.

Further Reading: Q. Ye, D. Doermann, "Text detection and recognition . . .," *IEEE PAMI* (37) #7, 2015.  
H. Lin, P. Yang, F. Zhang, "Review of scene text detection . . .," *ACM in Eng.* (27), 2019.

### Example 2.2: Pattern Recognition of the Mind

The retina in the eye is densely packed with light-sensitive cells, so that it may be tempting to think that our brain effectively sees and perceives the world as a great many pixels, much like the first feature in [Example 2.1](#). However there are many simple mind tricks or optical illusions that can make it clear that a systematic, pixellated view of the world is not really how we function.

With randomly fluctuating rotation, scale, and vertical offset ...

The typesetting may be weird, but for the human brain this is very easy to read,

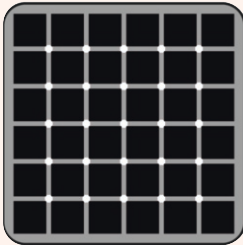
... but exceptionally difficult for a computer to do so.

Further evidence that our brain recognizes overall words or groups of words, rather than individual letters, stems from our ability to read text with permuted or missing letters:

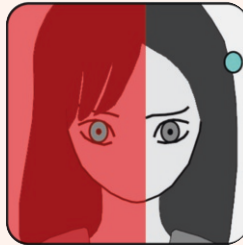
It deons't mettar in waht oerdr the lrttees in a wrod are,  
as lnog as the frist and lsat lltteres rmeain in the rght pacle.

\*t \*ls\* d\*\*sn't m\*tt\*r wh\*th\*r y\*\* h\*v\* \*ll \*f th\* l\*tt\*rs in \*v\*r\* w\*rd,  
\*s l\*ng \*s y\*\*r br\*\*n h\*s \*n\*\*gh \*nf\*rm\*t\*\*n t\* f\*ll \*n th\* g\*ps.

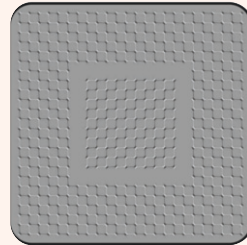
Indeed, the immense popularity of optical illusions in great part stems from us not being able to consciously understand why our brain's visual system is seeing a particular behaviour,



There are no black dots in the white circles.



The two eyes are the same colour. There is no blue pigment, at all, in the left eye.



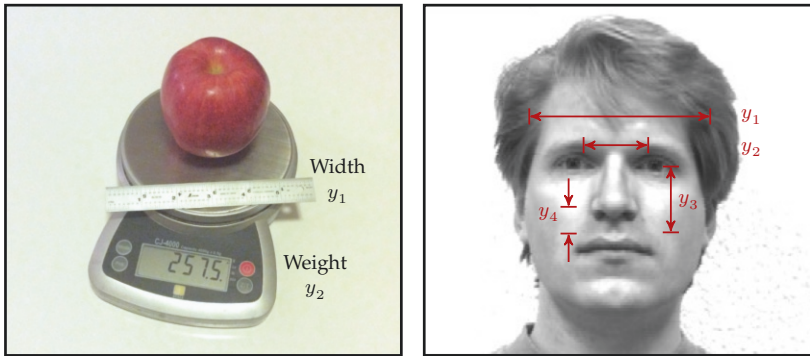
This is a static image, yet trying zooming in and scrolling.

... but which, among other reasons, stems from the fact that the eye is *not* just sending pixellated images to the brain. Rather, there is a great deal of *feature extraction* taking place, much already in the retina, which evolutionarily proved very helpful in running through a forest, but perhaps not so useful in staring at deliberately manipulated images on a page. To be fair, computer-based image recognition is susceptible to its own optical illusions ([Example 11.3](#)), in many cases far more primitive than those affecting the human visual system.

Further Reading:

D. Gershgorn, "Fooling the machine," *Popular Science*, 2016.  
D. Purves, R. Lotto, S. Nundy, "Why We See What We Do," *American Scientist* (90) #3, 2002.





**Fig. 2.2. MEASUREMENTS:** There is a wide variety of measurements which could conceivably be extracted for any object of interest, whether the size/weight of a piece of fruit, left, or various dimensions extracted from a face, right. (Right image from the Yale Face Database [2, 5]).

## 2.2 Measured Patterns

The word “pattern” may bring to mind texture, fabric, or shape. However in the context of *pattern recognition*, the notion of pattern is far more broad, and can apply to any *thing* that can be distinguished from another *thing*. Really, “identity” might have been a better choice of word: we would like to infer the unknown *identity* of an object, whether the type of wildflower, type of songbird, or the name of the person facing a camera — each of these has a certain identity which we would like to determine from measurements.

A pattern is assumed to have certain *properties* or *attributes* which distinguishes it from other patterns. One or more *measurements* are taken of a pattern, as shown in [Figure 2.2](#), which (should) reflect either directly or indirectly the attributes associated with the pattern. Indeed, one of the steps in pattern recognition is to assess to what extent a given measurement is relevant or helpful, as opposed to irrelevant or useless, to the classification task at hand. The selection of appropriate measurements is an essential part of the design stage of any pattern recognition solution development, since measurements may cost money and/or time, and poor measurements lead to poor performance of the resulting classifier.

As shown in [Figure 2.1](#), *features* are functions of the measurements,

$$\text{Measurements } \underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \xrightarrow{x = f(y)} \text{Features } \underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (2.1)$$

The process of transforming measurements into features is intended to somehow facilitate classification, normally in one or both of the following ways:

1. By reducing the dimensionality of the problem:  $n < m$

We need to develop a classifier in the  $n$ -dimensional feature space. Fewer dimensions are easier to visualize, easier to understand, are normally associated with learning fewer parameters, and will typically lead to more robust classifiers.

2. By creating features in which patterns are more clearly distinguished:

In any given problem, we may be constrained in terms of what sorts of measurements are available, perhaps on the basis of cost or the kinds of measurement instruments available. As a result, each measurement may only very weakly distinguish different pattern classes, and it may be that some function  $f()$  of the measurements can yield a feature which is much more discriminating.

It is important to understand that the feature extraction function  $f()$  can focus the information from  $\underline{x}$ , or it can remove irrelevant information from  $\underline{x}$ , but  $f()$  never *adds* information. This is known as the *Data Processing Theorem*:

*If  $\underline{x} = f(\underline{y})$ , then  $\underline{x}$  can never have more information than was present in  $\underline{y}$ .*

An effective feature extraction function  $f()$  can make the pattern recognition problem *easier*, however, in principle, the best possible classifier based on the measurements  $\underline{y}$  should perform at least as well as the best possible classifier based on the features  $\underline{x}$ .

Features may be intuitive or they may be quite abstract. Consider, for example, the measurements (with given units) which we might take of an electric motor:

$$\text{Measurements } \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \text{Motor voltage (Volts)} \\ \text{Motor current (Amperes)} \\ \text{Motor speed (RPM)} \\ \text{Motor winding temperature (}^\circ\text{C)} \\ \text{Surrounding air temperature (}^\circ\text{C)} \end{bmatrix} \quad (2.2)$$

then each of the individual measurements is clear, and a feature such as

$$x = f(\underline{y}) = y_1 \cdot y_2 \quad (2.3)$$

is easily intuitively understood as the motor power<sup>1</sup> in Watts, whereas another feature, such as

$$x = f(\underline{y}) = \sqrt{y_1 - y_2} - \frac{y_3}{y_4} \quad (2.4)$$

is uninterpretable, in units which do not make any physical sense, but which could perhaps be effective as a feature for classification.

It should be pointed out that (2.1) is not the *only* way of encoding the information regarding a pattern. There is a field known as *Syntactic Pattern Recognition*, in which patterns and features are understood more as grammatical/linguistic constructs, but which will not be considered in this text (although discussed briefly in [Chapter 13](#)).

## 2.3 Classes

The whole purpose of pattern recognition or classification<sup>2</sup> consists of assigning an object to a *class*. A class is a particular pattern, or possibly a group of patterns which are *similar* or *equivalent* in some sense. In a given problem, the set of classes  $\mathcal{C}$  is defined as

$$\mathcal{C} = \{C_1, C_2, \dots, C_K\}, \quad (2.5)$$

such that we have  $K$  different classes<sup>3</sup> from which to choose. Whether we know the classes, or what we know about the classes, or whether we even know the *number* of classes  $K$ , will depend on the kind of pattern recognition problem to be solved, as will be discussed in [Section 2.5](#).

The notion of a *class* presupposes some sense that members of a class share some common properties or attributes. However the definition of class can vary much be a function of the problem being solved, as is illustrated in [Example 2.3](#).

---

<sup>1</sup> Technically, for an AC (alternating current) motor, one needs to distinguish between instantaneous power, at an instant of time, and average power, integrated *over* time. This distinction need not concern us here.

<sup>2</sup> This text will consider the terms “recognition” and “classification” as equivalent.

<sup>3</sup> The careful reader will observe the inconsistency in identifying the number of classes as  $K$ , when the [textbook notation](#) indicates that scalars will be lower case, and upper case variables are used for matrices or abstract concepts, such as classes. However the use of  $K$  for the number of classes is exceptionally widely adopted in pattern recognition, so it seemed preferable to retain it here to avoid confusion.

### Example 2.3: What Is a Class?

A class captures the essence of a pattern, such that there can be many examples of the class, all of which are *similar* or *equivalent* in some sense. The definition of class is not inherent or absolute, rather it will depend on which problem is being solved.

So, for example, in a face recognition problem each person is their own class, so the set of classes would be defined as

$$\mathcal{C} = \{ \text{"Paul Fieguth"}, \text{"Bob"}, \text{"Jane"}, \text{"Ali"}, \dots \}$$

so that I would be a member of the "Paul Fieguth" class:



∈ "Paul Fieguth"

You might argue that there cannot be "many examples" of this class, as was suggested at the top of this discussion, since I am only one person. But in the face recognition problem there can in fact be *many* pictures of me, *all* of which should be associated with my class.

At a university we might define a different set of classes

$$\mathcal{C} = \{ \text{"Professors"}, \text{"Staff"}, \text{"Undergraduate Students"}, \text{"Graduate Students"} \}$$

such that

$$\text{"Paul Fieguth"} \in \text{"Professor"}$$

Such a set of class definitions might seem to make sense from an organization/categorization perspective, however in most cases there are not likely measurable or observable attributes that allow for categorization as undergraduate vs. graduate, or professor vs. staff, so it is probably actually not helpful to formulate a pattern recognition problem this way.

One *could* use pattern recognition to estimate age, in which case one might have classes like

$$\mathcal{C} = \{ \text{"0–10 years"}, \text{"10–20 years"}, \text{"20–30 years"}, \dots \}$$

such that now I appear in a class as

$$\text{"Paul Fieguth"} \in \text{"50–60 years"}$$

Although this is a legitimate pattern recognition problem, it is a bit artificial, in that really what we have is a continuous underlying value, *age*, which we have chosen to (somewhat arbitrarily) discretize into classes (as is discussed in further detail in [Example 8.1](#)). Really we do not actually have inherently distinct classes here, rather we should instead formulate this as a parameter estimation problem, such that continuous parameter *age* is estimated from measurements, as will be discussed in [Chapter 7](#).

Example continues ...

### Example 2.3: What Is a Class? (continued)

One case where classes *would* be inherent could, for example, appear in a taxonomy of life,

$$C = \{ \text{"Mammals"}, \text{"Reptiles"}, \text{"Amphibians"}, \text{"Fish"}, \text{"Sharks"}, \dots \}$$

such that

$$\text{"Paul Fieguth"} \in \text{"Mammals"}.$$

We will need some way of describing classes. Such descriptions largely fall into one of four types:

VIA PROTOTYPE, an idealized representation or notion of the “essence” of the class. The advantage is that each class is unambiguously defined, however with no scope for variability.

VIA PARAMETERIZED SHAPE, a generalization of the prototype, in which the class has a known shape (rectangular or elliptical, say), where the shape is described in some number of parameters (ellipse centre, rotation, and axis lengths, for example). The description is more flexible than that of a single prototype, but still requires the type of shape to be assumed or known.

VIA STATISTICAL DISTRIBUTION, such that we have some description of the likelihood or probability of a class member having a particular set of measurements or features. This approach is very comprehensive, however there will be circumstances when the statistics are not known and may be difficult to infer.

VIA SAMPLES, such that a set of given samples (many apples, or tigers, or bicycles) directly characterizes the class. When such samples are given the representation is highly convenient, since nothing further needs to be done to describe the class, however there may be storage and computational challenges, since all of the data need to be saved and then searched every time a classification needs to be undertaken.

There is some fluidity between these forms, in that we may try to infer a prototype or a distribution from given samples, or to generate typical samples from a given distribution. Furthermore, in principle the prototypes, distributions, and samples can all be defined both in the measurement space  $\underline{y} \in \mathbb{R}^m$  and in the feature space  $\underline{x} \in \mathbb{R}^n$ .

It should be clear that patterns do not need to be identical to belong to the same class: not all pictures of me (Example 2.3) are the same, or of tigers, or of apples.

There are at least two sources of variability present in the measurements associated with a single class:

1. The inherent variability within a class: Every class will consist of members which differ in some way. The degree and nature of the inherent variability will depend greatly on the class definition. So the “Fruit” class contains all manners of variability in colour, size, and shape; the “Apple” class is much more specific, but apples do come in different colours and patterning; the “Granny Smith Apple” class is even more specific, but still will have apples of different sizes or with more or fewer blemishes.
2. Noise or random variations<sup>4</sup> in measurement: Every measurement involves some sort of physical process which will be subject to error, such as thermal noise in electronics, or quantization noise in converting an analogue signal to a digital representation.

In many pattern recognition problems the class set  $\mathcal{C}$  is predefined and has been specified as part of the problem to be solved. Ideally, however, there would be significant value in putting thought into class definitions in the design stage of a problem, to avoid class definitions which are articulated poorly or vaguely.

## 2.4 Classification

A *classifier* is some function  $g()$ , possibly analytical (i.e., an equation) or a computer algorithm, which assigns a class label to a given feature:

$$g(\underline{x}) \in \mathcal{C} = \{C_1, C_2, \dots, C_K\} \quad (2.6)$$

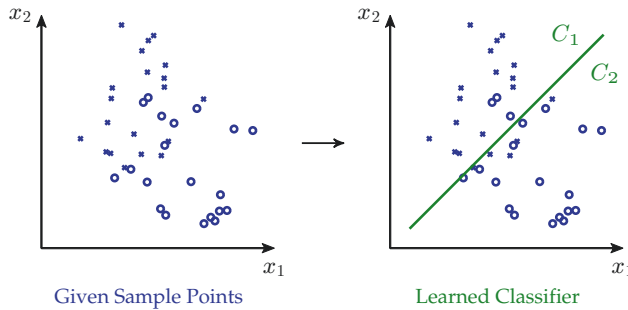
There is a strong relationship between feature extraction and classification, such that

- Good features allow for simpler classifiers, whereas
- Complex classifiers can compensate for weaker features, those features which are unable to fully separate the pattern classes.

A simple example of a classifier is illustrated in [Figure 2.3](#). In this case  $K = 2$ , meaning that the classifier is discriminating between only two classes. The classifier is a function of two-dimensional  $\underline{x} \in \mathbb{R}^2$ , and in [Figure 2.3](#) the classification boundary of  $g()$  is chosen to be a straight line, although *many* other classifiers are possible.

---

<sup>4</sup> *Noise* and *Randomness* are inherently statistical concepts, which are reviewed in [Appendix B](#).



**Fig. 2.3.** PATTERN RECOGNITION I — CLASSIFIER LEARNING: A classifier, here a straight line (right) dividing a feature space into classifications  $C_1$  and  $C_2$ , can be learned from a given set of sample points (left).

There are two fundamental steps associated with classification:

CLASSIFIER LEARNING is illustrated in [Figure 2.3](#). The method by which classifier  $g()$  is learned is normally a function of how classes are described. That is, we will see methods for learning  $g()$  on the basis of class prototypes, statistical distributions, or directly from sample data.

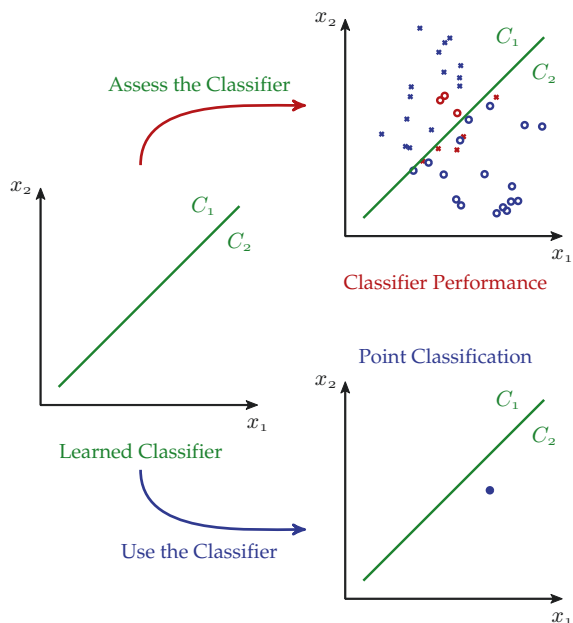
The whole topic of *Learning* is discussed in further detail in [Chapter 3](#), and specific types of classifiers will be discussed starting in [Chapter 6](#).

CLASSIFIER TESTING OR VALIDATION is illustrated in [Figure 2.4](#). Another important aspect of pattern recognition is the notion of testing, where we evaluate the performance of a classifier to assess how well it provides the correct class prediction. There are many ways to assess classifier performance; two of the most basic are

- Training accuracy: determine how well a classifier can assign correct classes to samples it was trained on, and
- Testing accuracy: determine how effectively a classifier can assign correct classes to test samples it has *not* been exposed to.

Classifier error will be assessed in detail in the statistical context, in [Chapter 8](#), and broader strategies for classifier *Validation and Testing* will be discussed further in [Chapter 9](#).

An important application of pattern recognition is for visual perception tasks, where the goal is to take an image (or video) as an input, and make predictions on the content of the image. Four illustrative examples are shown in [Figure 2.5](#), where we go from whole scene categorization and optical character recognition, to more complex visual perception problems such as object detection within an image and answering complex high-level questions about a scene.



**Fig. 2.4.** PATTERN RECOGNITION II — CLASSIFIER TESTING: What can we do with the learned classifier, left, from [Figure 2.3](#)? We could assess its performance (top), for example by counting how many sample points are classified correctly (blue) and incorrectly (red). Or we could apply the classifier (bottom) to a new, unknown point and then classify it.

## 2.5 Types of Classification Problems

With measurements, classes, and classification defined, the final step in introducing pattern recognition is to better understand what sorts of classification problems we might encounter. The range of problems is very large, from simple to complex, low dimensional to high dimensional, and given detailed models or training examples to being given almost no information at all.

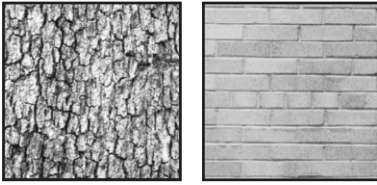
From this vast range of pattern recognition problems, we can most fundamentally group these problems into four scenarios on the basis of what is known and what is not known, as is illustrated in [Figure 2.6](#):

### SCENARIO 1: Model is known or given

This is the most information we can hope to have regarding a pattern recognition problem, in that we are told the behaviour of the measurements for each of the pattern classes. Normally this behaviour is characterized in



Texture Classification (Brodatz [3]):  
Classify each texture



Digit Recognition (MNIST [6]):  
Which digit is which?

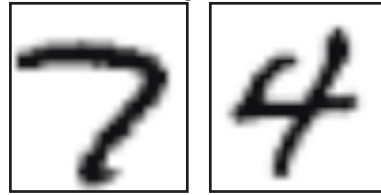


Image Segmentation (MS-COCO [4]):  
Find the airplane, the car, the bus ...



Image Segmentation (VisualQA [1]):  
Did the batter hit the ball?



**Fig. 2.5. PATTERN RECOGNITION ON IMAGES:** Since the human visual system is so dominant in human perception, a great deal of pattern recognition focuses on image-related problems. Here four examples are shown, from comparatively straightforward (top), the classification or recognition of whole images, to rather advanced (bottom), such as recognizing the objects within an image or being able to answer complex high-level questions.

a statistical<sup>5</sup> fashion, such as

$$p(y|C) = \text{The distribution of measurement vector } y \text{ given class } C \quad (2.7)$$

Such detailed information will be available only in those contexts where the physical process is known by which a given pattern class gives rise to measurements. It is very convenient to have such detailed information, since this allows statistical decision theory (Chapter 8) to explicitly define the optimal classifier, in the sense of minimizing the probability of classification error.

SCENARIO 2: Model is not known, labelled data are<sup>6</sup> available

Although we do not have an exact description of the problem, as in (2.7), we are given labelled data, meaning data pairs of the form

$$\{y_i, C_{\kappa}\} \longrightarrow \text{The } i\text{th measurement vector } y_i \text{ is known to have come from class } C_{\kappa} \quad (2.8)$$

<sup>5</sup> Statistical pattern recognition is discussed in Chapter 8, and the background mathematics on statistics and distributions can be found in Appendix B.

<sup>6</sup> I know it may look a bit odd, but in fact the word “data” is *plural*, so data are available. The singular is “datum”.

The four fundamental types of Pattern Recognition problems,

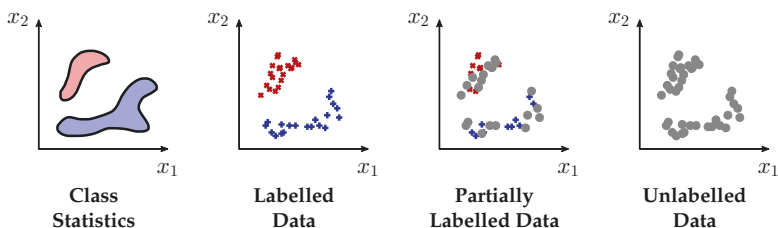
**Model-Based**

**Supervised**

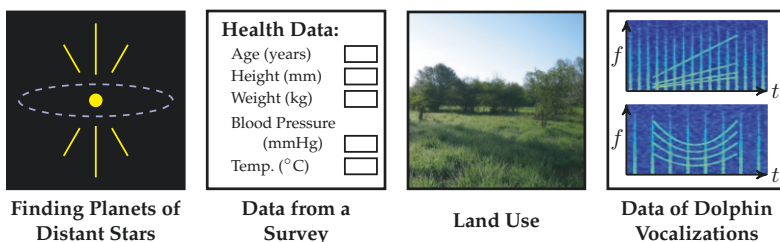
**Semi-Supervised**

**Unsupervised**

... based on what information you are given ...



One example of each ...



**Fig. 2.6. THE FUNDAMENTAL PROBLEMS:** There are four fundamental pattern recognition problems, ordered from the most detailed problem description (left) to the most ambiguous (right).

Labelled data do not just *magically* appear, of course; normally they have been labelled or tagged by a human observer, so we refer to this scenario as *supervised*, which can become exceptionally expensive or labour-intensive with larger datasets. Given labelled data, we have two broad approaches:

1. **Chapter 6:** We can derive a classifier directly from the given labelled measurements.
2. **Chapter 7:** We can learn an empirical probability model, as in (2.7), based on the labelled data, and then use the classifier which follows from the model.

SCENARIO 3: Model is not known, some data are labelled, some are not

This scenario would seem to be a trivial extension of Scenario 2, in that one could choose to learn a classifier (whether via **Chapter 6** or **Chapter 7**) on the basis of only those data points which are labelled. However such an approach *ignores* all of the *unlabelled* data; since labelled data can be expensive, requiring manual labelling, it is possible that *most* of the data will be unlabelled. This type of problem is referred to as semi-supervised, since some degree of human input is required.

Semi-supervised problems appear commonly in cases where we wish to use a small set of samples that have been manually labelled for classification (e.g., face images tagged by a human observer), but then also to leverage a very large set of unlabelled data. In actual fact, *most* pattern recognition problems are of this form, it is then more a question of whether the number of labelled data are sufficient for learning (back to Scenario 2), or not. Although semi-supervised learning is treated only very modestly, at the end of [Chapter 12](#), it builds very directly on all of the methods of inference, classification, and clustering, which are discussed in the rest of this text.

SCENARIO 4: Model is not known, no labelled data are available

Pattern measurements are available, however the points have no associated class information; this is known as an *unsupervised* problem. The range of problems here is still very broad, depending on whether we are told the *number* of classes, or their typical size or separation, or perhaps nothing at all. We refer to these as *clustering* problems, to be discussed further in [Chapter 12](#).

## Case Study 2: Biometrics

*Biometrics* are measurements that we can take of human characteristics. Most commonly we associate biometrics with the ability to validate who a person is, to uniquely recognize their identity. Common biometrics would include measurements of any of the following:

- Face
- Fingerprint
- Iris (the coloured region in your eye around the pupil)
- Retina (the pattern of arteries in the back of your eye)
- Veins (the vein structure in some part of your skin, normally the hand or arm)
- DNA

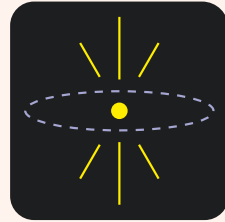
With the exception of DNA, where taking a measurement is much more invasive, all of the other biometrics on this list are based on extracting features from an image which can be acquired remotely (without touching the body).

To be sure, biometrics are not *only* about security and personal identification, and there is much that can be measured about a physical body that might not be helpful in recognizing an individual, but might be *quite* helpful in sports analytics, such as designing a superior golf club. However for the purpose of this case study our focus will be on the use of pattern recognition for uniquely identifying a person.

### Example 2.4: The Types of Pattern Recognition Problems

#### Scenario 1 — Model-Based: Planets around Distant Stars

Scientists are making claims about planets around distant stars, light-years away, and whether these planets are earth-like or not, having oxygen atmospheres. How is this even remotely possible?!



Scientists cannot actually *image* planetary atmospheres, let alone even see the planet itself at all. Instead, scientists have to *infer* the presence of a planet, its size, and its atmospheric constituents by having a *model* for the intensity of light, over time and as a function of wavelength, from that star. When a planet passes in front of the star, the star appears to dim slightly, and the planet's atmosphere contains molecules which influence the starlight at certain wavelengths. So we might have a vector of measurements, something like

$$[I_{\lambda_1}^b \ I_{\lambda_2}^b \ \dots \ I_{\lambda_q}^b \ I_{\lambda_1}^d \ I_{\lambda_2}^d \ \dots \ I_{\lambda_q}^d]$$

for intensity observations when the star is bright  $I^b$  and dimmed  $I^d$ , and at wavelengths  $\lambda_1, \dots, \lambda_q$  chosen to be at spectral lines of those atmospheric components of interest, such as water, oxygen, methane etc.

K. Heng, "A new window on alien atmospheres," *American Scientist* (105) #2, 2017

A second much simpler example could be a fetal heart monitor. Medical models suggest a normal fetal heart rate of 110–160 beats per minute. As a result, it would be easy to develop a monitor which sounds an alarm if the heart rate goes below 110 or above 160.

But where did such a model come from? Well, it came from a long history of observing the heart rates of many fetuses, and which of these were under some sort of distress or not. In other words, we gathered labelled data on heart rate (the measurement) and distress or not (the label), and then aggregated these data into a model.

So the model in a model-based approach could be derived from science or from labelled data — both approaches are legitimate.

#### Scenario 2 — Supervised: Surveys

We are exposed to nearly endless surveys (government census, political surveys, movie preference surveys, etc.) — every such survey represents *labelled data*, from which we can undertake supervised learning.

Suppose we wish to identify whether someone is a child or adult based on their height. What we can do is survey the heights (the measurement) of a large number of individuals, each person identified as "adult" or "child" (the label). Having each data point labelled is very convenient, because

Health Data:	
Age (years)	<input type="text"/>
Height (mm)	<input type="text"/>
Weight (kg)	<input type="text"/>
Blood Pressure (mmHg)	<input type="text"/>
Temp. (°C)	<input type="text"/>

Example continues ...

### Example 2.4: The Types of Pattern Recognition Problems (continued)

it allows us to know “truth” for each measurement, and to validate whether a proposed classifier is actually classifying correctly or not for each given measurement.

#### Scenario 3 — Semi-supervised: Land Use

With rare exceptions, nearly every pattern recognition problem begins as unlabelled, and the question then becomes how *much* labelling is needed for adequate learning, or how much learning can be afforded given a limited budget.

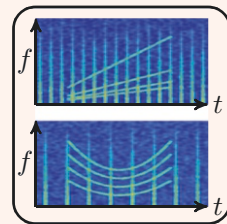


The illustrated example is that of land use: there are endless satellite maps and airplane photos of the earth’s surface (those are our *unlabelled* data), but sending out a team of people to actually do field surveys, on site (or perhaps trained operators sitting at a computer terminal, identifying houses and roads) costs money. That is the cost, however, of labelling the data: to actually *know* what land use (desert, city, agriculture, forest, water, swamp, . . .) is actually present at a given location. To be sure, the actual land-use categories (classes) could vary significantly from one context to another, depending on what you would like to study.

Although labelling may be expensive, even labelling only a modest number of points provides significantly more context than having no labelled data at all. At the same time, the *unlabelled* data points are also useful, since they may give us a much richer sense of class shape than the few labelled data points may be able to do.

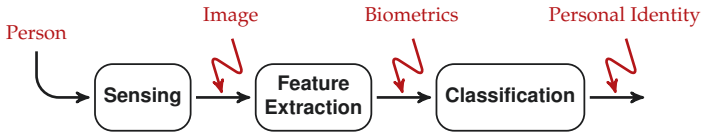
#### Scenario 4 — Un-supervised: Whale and Dolphin Vocalizations

Suppose we wish to identify the vocalizations of dolphins or whales, where the figure illustrates vocalization strength (colour) and frequency  $f$  as a function of time  $t$ .



We have no way of labelling the data, since there is no information available to us regarding the meaning or intent behind the vocalizations. Nevertheless there *are* patterns present in the data — there are particular sounds which are articulated, repeatedly, which could be represented as a cluster of points in some feature space. And there are *different* sounds that are made, each of which would appear as a *separate* cluster, assuming the feature space to have been appropriately chosen.

The pattern recognition challenge would then be a clustering problem ([Chapter 12](#)). The goal would be to look for naturally-occurring groups (each of which might be a word or a sentence), or possibly groups-of-groups, allowing us to begin understanding some aspect of the richness, complexity, or variations in dolphin and whale communications.



**Fig. 2.7. BIOMETRIC RECOGNITION:** In order to recognize a specific person on the basis of remotely-sensed biometrics, we need to acquire (sense) an image of some part of the body which has a unique signature (fingerprint, retina etc), extract biometric features from this image, and then develop a classifier that reliably recognizes the individual.

As we know from [Figure 2.1](#), which has been re-cast into the context of this case study in [Figure 2.7](#), a biometric-based human recognition system would have three basic components:

1. **Image Acquisition:** Outside the scope of this text, there are many design decisions related to imaging hardware, such as cost, ease of use, lighting, choices of sensor (infrared versus visible, grey-scale versus colour). Poor hardware, which yields images which vary significantly for the same person, places a greater burden on the subsequent steps of feature extraction and classification, so one goal is for the hardware to produce images as consistently repeatable as possible.
2. **Feature Extraction:** Also outside the scope of this text, given an image we now need to extract some vector of values which encode the behaviour observed in the image. Examples could include face geometry (as in [Figure 2.2](#)), a texture model of the iris, the locations and angles of arteries in the retina or lines in a fingerprint. This step is essentially in the domain of image processing: how to extract information from a given image.
3. **Classification:** Given a vector of features, the core component of the biometric system is a classifier, determining which person has just been measured.

The classifier returns one member of the class set

$$\mathcal{C} = \{NoMatch, Person_1, Person_2, \dots, Person_K\}. \quad (2.9)$$

It is important to have the *NoMatch* class present, otherwise the classifier is forced into classifying a given feature vector, even when there is little evidence that the feature vector is a good match. The reason why this is important is because of a significant asymmetry in the costs of two kinds of error:

$$\begin{aligned} \text{Classifying } \underline{x} \text{ as } NoMatch \text{ where } Person_1 \text{ is correct} &\leftarrow \text{Frustration} \\ \text{Classifying } \underline{x} \text{ as } Person_2 \text{ where } Person_1 \text{ is correct} &\leftarrow \text{Security Breach} \end{aligned} \quad (2.10)$$

Avoiding frustration and security breaches are not the only criteria at hand. In general, a successful biometric strategy must satisfy

- **Universality:** Every person should be measurable, regardless of age and health
- **Uniqueness:** The feature vectors extracted for a given person should be robustly unique
- **Consistency:** For a given person the feature vector should be highly repeatable from one try to the next, and should be slowly (or not at all) varying over time.

Clearly there are other criteria, such as non-invasiveness, social acceptability, or how easily the system would be to defeat via nefarious means.

How we would assess and validate the resulting classifier is a complicated topic which will be discussed at length later in this text ([Chapters 3, 4, 8, and 9](#)). In a nutshell, consistency is evaluated on the basis of class *spread* in feature space, uniqueness is assessed based on class *overlap*, and the two types of error in ([2.10](#)) would be evaluated based on test data (as discussed in [Chapter 9](#)).

### Lab 2: The Iris Dataset

The end of each chapter contains a worked numerical lab. The intent of the labs is to bring out some of the details in implementing pattern recognition concepts, so that the reader can see all of the steps laid out. The code is written in MATLAB, which was chosen because it is widely available at educational institutions, and because the code is fairly intuitive to anyone who programs in C or Python. Many tutorials can be found online, however no particularly detailed understanding of MATLAB is required to follow the lab discussions.

We have not yet learned any classifiers, but we can start by trying to understand what a pattern recognition dataset might look like. The *Iris Dataset* contains flower data for three types of Iris:

Iris Setosa



(Creative Commons, Tiia Monto)

Iris Versicolor



(Creative Commons, Danielle Langlois)

Iris Virginica



(Creative Commons, Public Domain)

Now certainly there is nothing profound about classifying these three particular plants, since there are many other types of Iris, many other types of flowering plants, and indeed many other pattern recognition problems of much greater significance. However the dataset is classic, having originally been published in 1936, it is relatively simple, and it is widely used in the pattern recognition field.

With reference to [Figure 2.1](#), we need to begin by identifying the classes and the measurements. There are three classes,

$$\mathcal{C} = \{C_1, C_2, C_3\} = \{\text{"Iris Setosa"}, \text{"Iris Versicolor"}, \text{"Iris Virginica"}\}$$

and for each plant four measurements were taken:

$$y = \begin{bmatrix} \text{Sepal Length} \\ \text{Sepal Width} \\ \text{Petal Length} \\ \text{Petal Width} \end{bmatrix}$$

This dataset and following code are available for you from the [the textbook data site](#):

```
load Iris
IrisData
```

which leads to the following output being generated:

```
IrisData =
```

```
    5.1000    3.5000    1.4000    0.2000    1.0000
    4.9000    3.0000    1.4000    0.2000    1.0000
    4.7000    3.2000    1.3000    0.2000    1.0000
```

... plus 147 more lines of data. Converting this into more human-readable form, what the data are saying is the following:

Sepal length	Sepal width	Petal length	Petal width	Class
5.1 cm	3.5 cm	1.4 cm	0.2 cm	Iris Setosa
4.9 cm	3.0 cm	1.4 cm	0.2 cm	Iris Setosa
4.7 cm	3.2 cm	1.3 cm	0.2 cm	Iris Setosa
⋮	⋮	⋮	⋮	⋮
5.5 cm	2.4 cm	3.7 cm	1.0 cm	Iris Versicolor
⋮	⋮	⋮	⋮	⋮
5.9 cm	3.0 cm	5.1 cm	1.8 cm	Iris Virginica



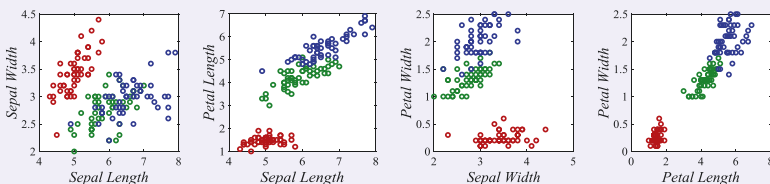
One of our first goals in a pattern recognition problem is to try to understand the nature of the problem:

- What are the class shapes? ... Round?, Elongated?, Split into several pieces?
- How localized are the patterns? ... Compact and small?, Spread out?
- How much do the classes overlap? ... Separated?, Touching?, Overlapping?
- How might we separate the classes? ... Straight line?, Curve?, Complicated?

Our first challenge is that we have four measurements, therefore each data point is a vector in four dimensions, which is quite difficult to visualize. What we *can* do is plot very nicely in *two* dimensions, so let us start by looking at a few 2D examples. In each case we are plotting sample points (circles) from each of the three classes (coloured) based on two of the measurements:

```
% loop over three classes and plot
clf
cols = 'rgb';
p=1;
for f = [1 2; 1 3; 2 4; 3 4]', % define the pairs of features to plot
    subplot(1,4,p)
    p = p + 1;
    for c=1:3,
        % plot the points associated with class in variable "c"
        q = find(IrisData(:,5)==c);
        plot(IrisData(q, f(1)), IrisData(q, f(2)),'o' cols(c));
        hold on
    end
    xlabel(IrisMeas(f(1)))
    ylabel(IrisMeas(f(2)))
end
end
```

The resulting plots then look like



Class points: Iris Setosa Iris Versicolor Iris Virginica

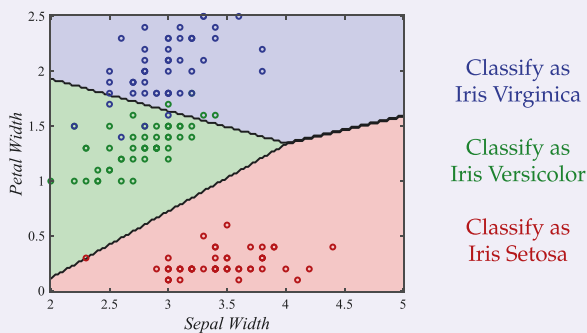
What do we observe here?

1. Iris Setosa (red) is, in general, quite easily separated from Iris Versicolor and Iris Virginica, which overlap in each of the pairs of measurements which we plotted.
2. Class compactness and degree of overlap vary from one measurement to another. The sepal data (leftmost plot) are least compact and most overlapping, and the petal data (rightmost plot) is more compact, and only slightly overlapping.
3. Generally the classes are compact. It is primarily the sepal width measurement (first and third plots) in which the classes exhibit outliers (data points somewhat apart from the rest of the class).
4. The visualization is necessarily incomplete. If the classes overlap in 2D, you *cannot* tell from the 2D plots whether the classes overlap or not in 4D. Ideally we would be motivated to extract two or three features from the four measurements, to allow for better visualization.

A classifier is a function  $g()$ ,

$$g(\mathbf{y}) \in \mathcal{C} = \{\text{"Iris Setosa"}, \text{"Iris Versicolor"}, \text{"Iris Virginica"}\} \quad (2.11)$$

which maps every possible 4D measurement to one of the three classes. The classifier is not "required" to actually use all four of the measurements so, for example, we might express the classifier as a function of two measurements,  $y_2$  and  $y_4$ :



We see that the classifier does not classify all points correctly — there are green data points in the blue domain, and blue points in the green

domain. The classifier is also not optimal — it would be easy to adjust the classification boundaries slightly to improve the performance. How to deduce such a classifier, and the options we have available to us in selecting a classifier, will be one of the central themes in this book.

## Further Reading

The [references](#) may be found at the end of each chapter. Also note that the [textbook further reading page](#) maintains updated references and links.

Wikipedia Links: [Pattern](#), [Pattern recognition](#) [Biometrics](#)

The most fundamental background on the algebra, statistics, and optimization theory needed to understand this book are discussed in [Appendix A](#) through [Appendix C](#).

The concepts in this chapter span much of pattern recognition, but were deliberately discussed only very briefly. Really the entire rest of this textbook represents further reading to the ideas introduced in this chapter.

## Sample Problems

### Problem 2.1: Short Answer

In your own words, give a short definition of each of the following:

- Data Processing Theorem
- Feature extraction
- Classification

### Problem 2.2: Short Answer

Given an example, different from the ones given in this chapter, for each of the four fundamental types of Pattern Recognition problems:

- Model-Based
- Supervised
- Semi-Supervised
- Unsupervised

**Problem 2.3: Lab/Computational**

**Lab 2** code and associated Iris data are available to you from the [the textbook data site](#).

- (a) Using the code from the website, or by writing your own, reproduce the plot on [page 25](#).
- (b) Generalize the figure by plotting *three* of the features in a 3D plot.
- (c) Rotate the 3D plot of part (b) to allow the three classes to be maximally distinguished. What are you actually doing, in rotating the plot this way?

**Problem 2.4: Reading — Semi-Supervision**

The problem of semi-supervision seems to be located half way between the supervised labelled-data problem and the unsupervised unlabelled-data problem. In principle we might imagine solving it as

1. A labelled-data problem, progressively classifying the unlabelled points to their closest labelled counterparts, or
2. An unlabelled-data problem, performing unsupervised clustering on the data points.

Look up semi-supervision and offer a summary of the prevailing strategies.

**References**

1. S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Zitnick, D. Parikh, Vqa: Visual question answering, in *International Conference on Computer Vision (ICCV)* (2015)
2. P. Bellhumer, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(7), 711 (1997)
3. P. Brodatz, *Textures: A Photographic Album for Artists and Designers* (Dover, 1966)
4. T. Lin et al., Microsoft coco: Common objects in context. arXiv:1405.0312v3 (2015)
5. A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 643 (2001)
6. Y. LeCun, C. Cortes, C. Burges, *The MNIST Database of Handwritten Digits*