AI and Machine Learning: The Basics

Nicolas Duchateau, Esther Puyol-Antón, Bram Ruijsink, Andrew King

DOI = https://doi.org/10.1007/978-3-031-05071-8_2

2 AI and Machine Learning: the Basics

Dr Nicolas Duchateau[a,b,*]

Dr Esther Puyol-Antón[c]

Dr Bram Ruijsink[c,d]

Dr Andrew King[c]

[a] *Univ Lyon, Université Claude Bernard Lyon 1, INSA-Lyon, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69621, Lyon, France.*

[b] *Institut Universitaire de France (IUF), France.*

[c] *School of Biomedical Engineering and Imaging Sciences, King's College London, United Kingdom.*

[d] *Heart and Lungs Division, University Medical Centre, Utrecht, Netherlands.*

[*] *Corresponding author.*

Authors' contribution:

- Main chapter: ND, BR, AK.
- Tutorial: EPA, ND.

Abstract

In this chapter the key concepts of artificial intelligence and machine learning are introduced. The importance of first identifying and defining the right problem is emphasised. A review is provided of different types of machine learning model, and pointers are provided about how to design and train a model to meet the requirements of the chosen problem. Important considerations regarding validating the trained model are also discussed. A review is provided of the context of AI and machine learning in cardiology, i.e. what imaging and non-imaging data sources are typically available for such models and what information can they provide? Within each of these data sources, some of the important applications and contributions of AI are highlighted. A practical tutorial is provided to introduce the reader to Jupyter notebooks and Python.

Keywords:

artificial intelligence, machine learning, data descriptors, data standardization, validation, echocardiography, magnetic resonance, computed tomography, positron emission tomography, electrocardiogram, electronic health records

Learning Objectives:

At the end of this chapter you should be able to:

- *O2.A Clearly define the right problem and justify why machine learning is needed to solve it*
- *O2.B Describe the different classes of machine learning model and in what types of situation they can be applied in*
- *O2.C Outline a design for a machine learning model to address a given problem in a given medical scenario*
- *O2.D Describe how machine learning models can be fairly and quantitatively validated*
- *O2.E Describe the main sources of data for machine learning models in cardiology*

Introduction

In this chapter we will delve into the world of AI and machine learning in a bit more detail. We will look at what issues we need to consider and what decisions we should make when looking to develop a machine learning model to address a specific problem. We focus on machine learning in general, but everything that we write is also applicable to the specific field of deep learning[11]. The chapter closes with some exercises intended to reinforce what has been learnt, as well as the first of our practical tutorials, which is a chance for you to "get your hands dirty" by starting to do some simple programming using Python and Jupyter. This tutorial acts a groundwork for the more specific tutorials on different topics that will be presented in future chapters.

Defining the Problem

As well as curating a database for training our AI model, it is important to think about and clearly define which problem we want to address. For example, our problem could be the diagnosis of a disease, the characterization of the function of an organ, or simply the anatomical alignment of two or more medical images. Identifying and clearly defining the problem is an essential step - as we discussed in the previous chapter, the details of which annotations (if any) we add to our data depends upon our problem. The way in which we define our problem also impacts upon which AI model(s) can be used to address it, as we will see in the next section.

Key considerations in defining the problem are the role of the AI model in the clinical workflow, as well as the potential risks involved in incorporating it. For example, if we want our model to diagnose a disease that is normally diagnosed by a radiologist, do we want to replace the radiologist or assist the radiologist by automating 'obvious' diagnoses whilst flagging up 'difficult' ones for manual review? If we aim to identify potential disease at an earlier stage, what would happen to patients who are identified in this way? Do effective treatments exist? How invasive are they and does their benefit

[11] We introduce the technical aspects of deep learning in Chapter 3.

outweigh their risk? Such considerations are often overlooked when proposing AI models in medicine, and we revisit this important topic in Chapter 9.

Types of Model

Once the problem has been clearly defined and we are sure that there is a beneficial role for AI to play, we can start to think about which model to employ. Focusing now specifically on machine learning techniques, it is normal to break down types of model into two main classes:

- *Supervised models*: The aim of a supervised model is to predict an output given an input. To train a supervised model it must be provided with a database of input/output pairs, and typically the outputs are produced by annotating the database. For example, to revisit our disease diagnosis problem, in this case the inputs would be medical images such as MR or CT scans, and the outputs would be binary labels (i.e. disease/no disease).
- *Unsupervised models*: With unsupervised models no output label is used. The aim of the machine learning model is to analyse the input data (e.g. images) and try to uncover patterns that might be useful for subsequent processing. These patterns can be as simple as identifying 'clusters' of similar inputs, or they can be more sophisticated representations of relations between inputs, as we will see below. The reason for not using labels could be that they are not available, that they are insufficiently trusted (e.g. distinguishing normal and reduced ejection fraction may be too reductive against the spectrum of heart failure [193]) or that supervised formulations showed their limits [100].

Because annotation can be a time-consuming process, we are often in the situation where we only have annotations for a subset of the training database. In such cases, rather than using supervised learning on the smaller subset, a class of techniques known as *semi-supervised learning* [78] can be employed. These techniques are able to exploit both the annotated and unannotated data to produce a model with better performance.

A third class of machine learning models, which has been less widely used in medicine so far, is *reinforcement learning*. Reinforcement learning techniques are neither supervised nor unsupervised. To understand the way in which reinforcement learning works, consider a toy problem of a mouse trying to

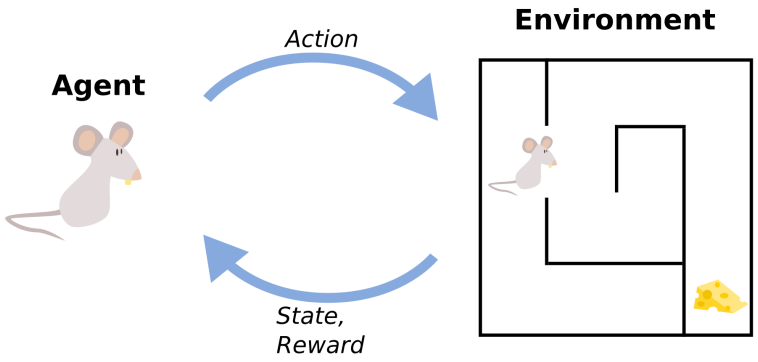


Figure 2.1: Reinforcement learning. An agent continually chooses an action which results in a reward as well as a new state in the environment.

navigate a maze to find a piece of cheese (see Figure 2.1). An AI *agent* is defined, which always has a current *state* in the *environment*. For example, the mouse agent will always have a location in the maze. To train the mouse, it will choose an action (a direction in the maze) which will result in a new state (location) as well as a *reward*. Good actions (i.e. those which result in eventually getting the cheese) are rewarded and bad actions are punished. The idea is that, by trying to solve the problem enough times and being rewarded/punished for its actions, the agent will learn to choose good actions. Although seemingly an abstract concept, applications in medicine have been proposed, for example in learning sampling strategies in MR [309], operator guidance in ultrasound imaging [267] and personalising computational models [279].

The choice of which class of machine learning model to employ depends upon what type of problem we have. If our problem can be clearly defined in terms of inputs and known and trusted output labels, then supervised learning can be employed. If a larger amount of extra unannotated inputs are available, semi-supervised learning can be considered. If no output labels are available or they are not sufficiently trusted, and the aim is simply to learn about the structure and patterns in the input data, then unsupervised learning should be used. Finally, if the problem can be formulated in terms of actions, states and rewards, then reinforcement learning can be investigated.

After the problem has been analysed and an appropriate class of technique has been identified, a specific machine learning model must be chosen. There

Supervised learning	
Classification	Regression
Linear discriminant analysis (LDA) *Support vector machines (SVM)* *Logistic regression* *Decision trees/forests* *Genetic algorithms* *Neural networks*	*Linear regression* *Ridge/kernel regression* *Decision trees/forests* *Genetic algorithms* *Neural networks*
Unsupervised learning	
Clustering	Dimensionality reduction
K-means *Mean shift* *Expectation maximization* *Hierarchical*	*Principal component analysis (PCA)* *Independent component analysis (ICA)* *Manifold learning*

Figure 2.2: Examples of machine learning models broken down by class.

has been a wide range of models proposed over the years for supervised and unsupervised learning. In Figure 2.2 we summarize some of the more commonly used ones. For a more detailed review and specific references we recommend [322].

We can see that supervised learning models can be broken down further into *classification* and *regression* methods. The distinction here lies simply in what type of output we want to estimate. If the output type is categorical or ranked (see Figure 2.3), then a classification model must be used. If the output type is discrete or continuous then a regression model must be used. For example, in our disease diagnosis example the output label (disease/no disease) is binary and categorical, so a classification model would be appropriate. Similarly, the segmentation of anatomical structures involves assigning a category to each pixel of an image, and can be seen as a (pixelwise) classification problem. On the other hand, estimating a numerical biomarker directly from an image or set of images, such as left ventricular ejection fraction (EF) in cardiac imaging, would require a regression model.

Unsupervised models can be broken down into *clustering* and *dimensionality reduction* methods. With clustering the aim is to identify a limited number of groups of inputs that are similar in some way, i.e. they represent clusters in the distribution of inputs. In dimensionality reduction, the input data

Categorical (a.k.a. nominal)	Ranked (a.k.a. ordinal)	Discrete	Continuous
Non-quantitative No ordering	Non-quantitative, Meaningful ordering of values	Quantitative, Can take a limited number of values	Quantitative, Can take any value within a range
E.g. blood type	*E.g. tumour grade (grade I, grade II, etc.)*	*E.g. number of tumours*	*E.g. blood pressure*

Figure 2.3: A summary of statistical types of data.

are mapped, or transformed to a new coordinate system, in which further analysis can take place. Standard techniques for this include the use of linear (principal component analysis - PCA) or nonlinear (manifold learning) transformations.

Model Design

Having considered the type of machine learning model we can employ, we now move on to a range of other design considerations, mostly related to the data used to train and evaluate the model.

Data Descriptors

As for clinical observations and standard statistical analyses, choosing adequate inputs is key for effectively training a machine learning model. A data *descriptor*, also referred to as a *feature*, summarizes the information available in each of the studied samples. The traditional machine learning paradigm generally dissociates the feature selection and problem solving tasks. This means that the machine learning developer relies on prior knowledge of the application area to select one or several features, which are then used as inputs to the model during training and evaluation. This process is known as *hand-crafting* of features or descriptors. In contrast, in deep learning a feature representation (based on 'raw' inputs provided by the user) that is optimized for the problem being addressed is learnt from the data and used to solve the problem. It therefore stands as a powerful tool for new discoveries from the

data, although this often comes with the cost of reduced interpretability[12] and control that a hand-crafted feature set could provide.

The simplest type of feature consists of single values, also called scalar measurements. These can be previously extracted from images such as cardiac chamber dimensions or EF, or correspond to more advanced image characteristics at the pixel level, such as radiomics features [137]. They can also be measured by other means (such as pressures or brain natriuretic peptide (BNP) levels) or even correspond to patient characteristics or external factors.

In the case of scalar input features, machine learning stands as a way to model more complex associations between the input features (and output labels, if any) than standard statistical approaches. However, inputs can also consist of more complex data structures such as signals or images, or even descriptors extracted at each location of these signals or images. The complexity of such descriptors is quantified by their *dimensionality*[13]. Nonetheless, the *intrinsic dimensionality* of these descriptors is generally much lower than the dimensionality of the data: the intrinsic dimensionality is the actual number of degrees of freedom that govern the *observed* data. Dimensionality reduction techniques from the field of representation learning [49, 423] provide an approximation of this intrinsic dimensionality, and a simplified representation of the data that can be used as a new input for the machine learning model. Figure 2.4 illustrates these considerations for the study of myocardial deformation from cardiac imaging data, using a single scalar value at each American Heart Association (AHA) segment or more complex descriptors at each point of the left ventricular myocardium.

Using several input descriptors is rather straightforward for scalar measurements, which can be considered as elements of a higher-dimensional vector that concatenates them (after they have been normalized). In contrast, combining several high-dimensional descriptors of potentially heterogeneous types is an ongoing field of research, addressed both with machine learning

[12] *Interpretability* refers to the ability of humans (e.g. end-users or model developers) to understand the process by which a machine learning model arrived at its output based on the input data. We deal with model interpretability in more detail in Chapter 8 (page 205) and Chapter 9 (page 227).

[13] Dimensionality of data refers to the number of degrees of freedom they have, for example 10^4 for a two-dimensional (2-D) image made of 100×100 pixels.

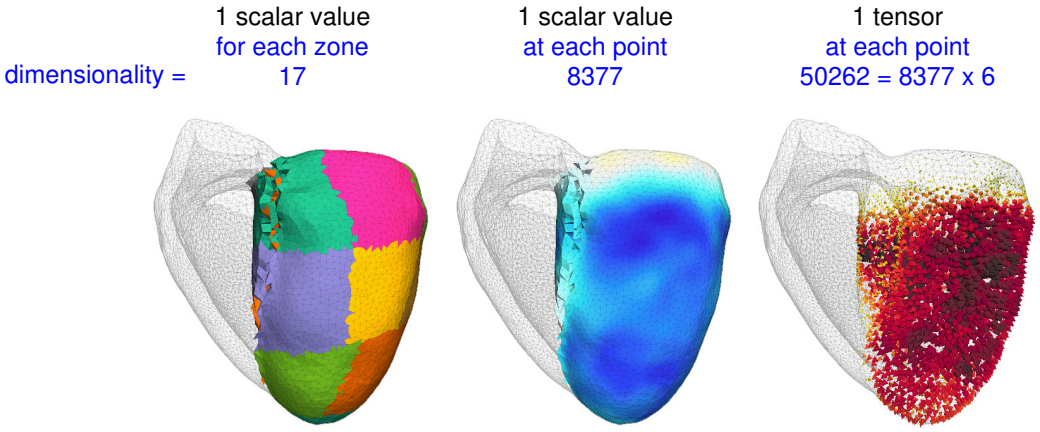


Figure 2.4: Different choices of data descriptor for myocardial deformation (strain) on a three-dimensional mesh of the heart's left ventricle and their associated dimensionality.

or deep learning algorithms [230] and will be further discussed in Chapter 8.

Data Constraints

Working with medical data requires specific care to preserve the properties of the data descriptors, and guarantee the soundness of observations.

A first example from cardiac imaging may help for understanding: let us consider again the analysis of myocardial deformation using strain data. In one dimension, strain is a scalar that represents the relative change in length of an object with respect to a reference state, typically between end-diastole and end-systole (Lagrangian strain). In three dimensions, strain quantifies the deformation of a three-dimensional (3-D) object (e.g. a cube representing a small portion of the myocardium at a given location), and is represented by a 3×3 *tensor* (a symmetric matrix that belongs to a specific family of matrices). This means that strain is no longer represented by a single scalar value but by 6 matrix coefficients (because the matrix is symmetric). It also means that standard operations such as addition, multiplication and averaging across a population may not preserve the tensor properties of the strain descriptor, and may result in physiologically implausible results.

A second example, also from cardiac imaging, complements this view on the

allowed operations on such descriptors. Consider a dataset of segmented acute myocardial infarcts, from the same coronary territory. Estimating a representative infarct pattern across a subgroup of subjects may be highly informative. Nonetheless, computing the linear average of several binary infarct patterns (previously aligned to a common reference, see the next section) results in a non-binary pattern with intermediate values that no longer resembles a plausible infarct. In this case, the machine learning model needs to consider the nonlinear structure of the space of infarct patterns so that the analysis always corresponds to plausible infarct patterns (Figure 2.5). In general, when choosing data descriptors for use by machine learning models, one should be aware of these limitations and decide the level of approximation that can be tolerated on the computations and results, and adapt the learning algorithms accordingly.

To return to our myocardial strain example, this means that one can decide to work with (see Figure 2.4):

- A single scalar value that summarizes myocardial deformation, such as strain in a given direction, at a given instant and averaged over the myocardium (e.g. peak global longitudinal strain). Here, standard comparisons between values are allowed.
- A high-dimensional object that encodes strain in a given direction, but for several instants in the cardiac cycle and/or several locations across the myocardium. Here, the model may consider each temporal instant or spatial location independently from the others, or find metrics or data representations that take into account the spatiotemporal consistency of these patterns, such as dimensionality reduction techniques.
- A strain tensor at several instants in the cardiac cycle and/or several locations across the myocardium. Here, the model should also preserve the properties of such tensors, often addressed with specific metrics and nonlinear operations [302].

Naturally, these choices are conditioned by the complexity of the question to be addressed, the amount of samples available (as more complex descriptors/questions/models require larger populations) and the risk associated to the approximations made.

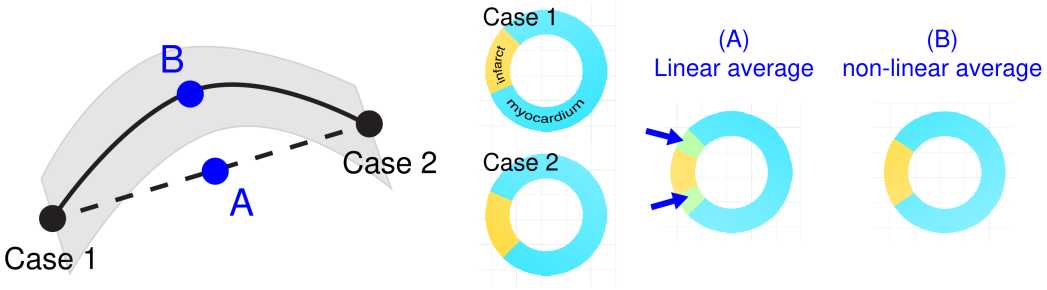


Figure 2.5: Linear and nonlinear average of two synthetic binary infarct patterns. As the space of infarct patterns is nonlinear, the average of two cases lies out of this space and does not correspond to a plausible infarct pattern (if the pixel labels for myocardium and infarct respectively correspond to 0 and 1, intermediate values of 0.5 are observed around the infarct zone shared between the two cases, as pointed out by the blue arrows). Machine learning models that handle this type of data should also consider potential nonlinearities in the data space to prevent bias in the analysis.

Data Standardization

If descriptors of heterogeneous types are used as inputs for learning, standardization of their values may be required to prevent imbalanced contributions due to incompatible units or scaling.

As noted earlier, several scalar descriptors can be concatenated to form a new one of higher dimensionality, but it is important to remember that they should be preprocessed so that their minimum/maximum values or their average/variance values match. Specific algorithms may require binarizing or categorizing the descriptors, or more advanced schemes such as *one hot encoding*[14]. A detailed list of normalization operations can be found in many standard machine learning libraries[15].

For high-dimensional descriptors of heterogeneous types, preprocessing may consist of finding a new representation of the data where more standard average/variance normalization can be achieved, using dimensionality reduction techniques such as linear PCA or nonlinear manifold learning. Among

[14] One hot encoding refers to the binarization of categorical data, resulting in a sequence of binary values, one for each category, in which only one value is equal to 1.

[15] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

nonlinear techniques, one interesting standardization approach consists of replacing the input descriptors by affinity matrices that encode the similarities between pairs of samples, generally achieved by (Gaussian) kernel functions [90].

Finally, prior standardization of the descriptors may be required to lower the effect of anatomical and timing differences between subjects. These techniques belong to the field of computational anatomy [265, 266] and statistical atlases [429], which is under active research. Nonetheless, well established techniques already provide acceptable reference systems of coordinates to which each subject's data can be transported. Popular methods consist of Procrustes alignment [144], registration, or parameterization techniques to estimate inter-subject correspondences and a reference anatomy, followed by interpolation or parallel transport of the subject-specific data to this reference. Temporal alignment may consist of interpolation based on physiological events [306], dynamic time warping [337] or temporal registration.

Model Validation

Training a machine learning model means that the model parameters are optimized to solve the targeted problem on a given dataset (the training set). However, the actual challenge of machine learning is to guarantee enough model performance on new samples not used for training, also referred to as the *generalization ability* of the model. Otherwise, the model would be *overfitted* to the training data and therefore be of less practical use.

In the example introduced above of diagnosing a disease from a medical image, which can be seen as a supervised classification problem, the training set samples consist of pairs of images and diagnosis labels that serve as ground truth to guide the optimization process. During training, the optimization process determines the model parameters (e.g. the logistic regression coefficients, or the neuron weights) that lead to the best classification on the training set (potentially balanced by some regularization that we will discuss later on). Then, the optimized model is applied to new images not necessarily from the same study, from the same institution and/or acquired with the same device, etc. This new set of samples is referred to as the *testing set*, and the model is expected to show a comparable classification performance on this new dataset, which would validate its relevance.

In general, the machine learning developer should prepare three different datasets:

- The *training* set, which is used for optimizing the model parameters.
- The *validation* set, which is used to evaluate the performance of the trained model on new samples not used for optimizing the model parameters[16].
- The *testing* set, which consists of the actual data to analyze with a previously validated model.

The validation set is different from the training set, and therefore is not used to optimize the model parameters. However, it may be used for selection of optimal *hyperparameters* of the model (external values that control the model behavior, which are fixed during training). This can be seen as a complementary training of the model.

The validation set may consist of samples from another study, and in this case the validation procedure is referred to as *external validation*. However, in practice, *internal validation* is generally performed: the validation set consists of a subset of the training set. A more robust evaluation is obtained by repeating this procedure several times and averaging the performance results. A typical scheme for this consists in partitioning the training set into blocks, and each time perform the validation on a different block. This procedure is known as *k-fold cross validation* when validation is repeated on k different blocks, or *leave-one-out cross validation* when a single sample is left out for validation (the remaining samples being used for training), the process being repeated to cover all samples.

The validation of supervised learning models provides two types of measures: the model performance on the training set, also called *bias*, and its performance on the validation set, also called *variance*. A non-optimized or wrong model would result in a large error on both datasets and would *underfit* the data, resulting in a high bias. Conversely, a model may *overfit* the training data and therefore perform poorly on the validation data, resulting in a low bias but a high variance. A validated model should therefore propose a trade-off between bias and variance, so that it generalizes well to the new

[16] But in some cases they can be used during training, e.g. for deciding when to terminate an iterative optimization process such as that used in training artificial neural networks, see Chapter 3.

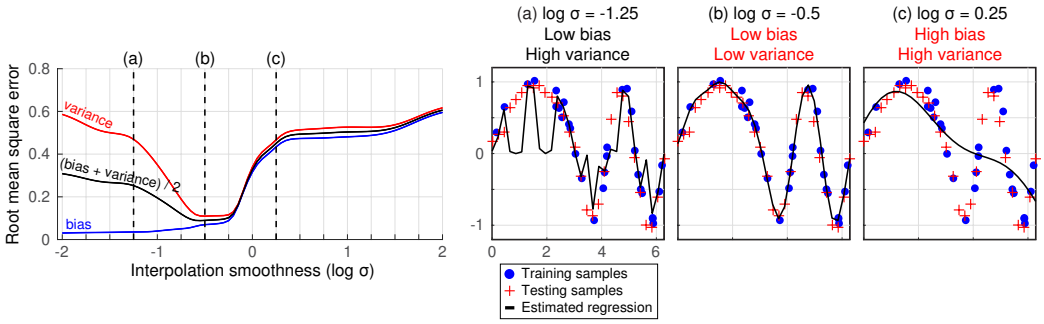


Figure 2.6: Determining the optimal model through bias and variance curves. Example on a nonlinear regression model with a hyperparameter σ that controls the smoothness of the regression, namely the simplicity of the fit to the training data.

samples from the testing set (see Figure 2.6).

Standard metrics for assessing the performance of supervised models consist of error measurements that are problem-specific:

- For segmentation, the Dice coefficient (overlap between segmentation and ground truth) or the Hausdorff distance (maximal distance between segmentation and ground truth boundaries), etc.
- For classification, measures derived from the amount of well predicted (true positive and true negative) and mispredicted (false positive and false negative) samples, such as sensitivity and specificity, or precision and recall, the area under the ROC curve[17], etc.
- For regression, errors on the predicted values such as the sum-of-squared differences or the root mean square error, etc.

The validation of unsupervised learning models is more challenging as labels are not available or used, and the user should find alternative ways of justifying the generalization of the model:

[17]The *receiver operating characteristic* (ROC) curve is used when assessing performance in situations where we have predicted and ground truth binary labels (e.g. disease classification). The ROC curve plots sensitivity against one-minus-specificity, for different predictor threshold values. The *area under the receiver operating characteristic curve* (AUC) is another measure of performance and is equal to the area under the ROC curve. AUC values range between 0 and 1 with 1 indicating perfect performance. See https://en.wikipedia.org/wiki/Receiver_operating_characteristic for further details.

- For clustering, the separability of the estimated clusters, their consistency across different datasets or different parameters, etc.
- For dimensionality reduction, the proportion of dimensions that explain most of the data (the model *compactness*), the realism/relevance of samples generated from the low-dimensional representation (the model specificity), etc.

For the sake of fairness, these metrics should differ from the measures that are minimized during the model optimization.

One can easily appreciate that better generalizability of the model can be achieved from the data perspective by increasing diversity in the training set, and from the model perspective by improving the model while balancing the adherence to the training samples, so that the validation samples are also well modelled. This last process is achieved by adding *regularization* constraints to the model, for example ensuring that the regression trend or the classification border are smooth, or that the segmented structures have smooth boundaries.

We encourage the reader to carefully consider these aspects, which are key for deploying a model on new cohorts and having a fair estimation of its relevance. Testing state-of-the-art algorithms on different datasets or applications is a good start: the more variety in the data, the higher will be the clinical trust in the model generalizability. Starting with simple models is highly recommended, as they may have lower performance compared to more sophisticated models but they can often generalize better to new samples.

Machine Learning is not a Panacea!

We would like to remind both starting and experienced developers that machine learning tools are actually *models applied to data*. By the principle of Occam's razor, a model should be as simple as possible whilst still enabling the problem to be addressed satisfactorily. Model complexity is related to the number of parameters in the model (e.g. for deep learning methods: the neurons' weights and the hyperparameters that govern the global behavior of the model). One should therefore start by carefully looking at the available data, using simple descriptors and simple models (including standard statistical methods), carefully test state-of-the-art methods on their own data,

and then decide whether the complexity of the question and the amount/-diversity of samples warrant investigation of more advanced models or data descriptors. In short: start simple!

But naturally, the model performance is only as good as the data. One may not expect stunning results on testing data that differ significantly from the training data, or with different data quality and/or confidence in the labels. Data are produced and curated by humans, so machine learning models are subject to the same biases and prejudices as humans. Furthermore, some machine learning models (e.g. deep learning) work best when trained with a lot of data, and such datasets can often be difficult to curate. In this context, recent research on model interpretability [276] (see Chapter 8, page 205) and uncertainty [133] (see Chapter 5, page 128) are certainly promising areas to explore to complement model validation in the near future.

Sources of Data for Machine Learning in Cardiology

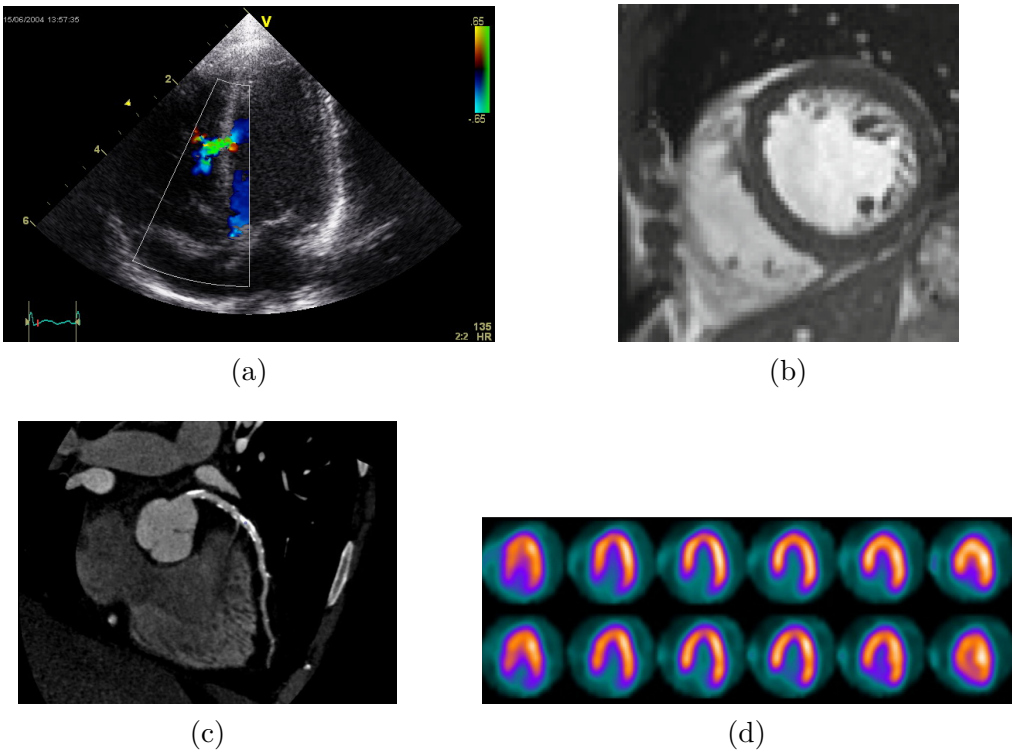
We have referred several times already to the importance of data, both in terms of data (and annotation) quality and the amount of data available. In this section, we review the data sources that are commonly available for training and validating machine learning models in cardiology.

Care of patients in cardiology relies heavily on data. During initial assessment as well as follow-up, detailed descriptions of data related to the patient's disease is recorded in health records. This includes a description of a patient's history, the symptoms he/she experienced, findings during physical examination, biophysical measures (heart rate, blood pressure etc.), additional testing results (electrocardiograms, biochemistry, imaging etc.) and finally treatments that have been administered. In principle, all of these data can be exploited by machine learning models, although the extent to which these possibilities have been explored varies.

In the last decade, most hospitals have implemented electronic health record (EHR) systems, allowing patient data to be stored in a systematic way. Raw image data from imaging exams are usually not stored in the EHR itself. Because of their size, the imaging exams are usually stored in separate, dedicated image storage systems (PACS: picture archiving and communication systems). The medical data stored in EHR and PACS systems forms a valuable resource for large data-driven studies in cardiology and other fields in

medicine. Below, we will review the most widely used imaging and non-imaging data sources in cardiology, and briefly discuss their technical background, practical use and place within clinical care. Brief summaries of machine learning models based upon these data sources will be provided.

Imaging Sources



(a) (b)

(c) (d)

Figure 2.7: Examples of cardiac imaging modalities: (a) echocardiography (with blood flow velocity measured using Doppler imaging shown as a colour overlay), (b) a frame from a cine CMR acquisition, (c) coronary CT angiograph, and (d) myocardial perfusion SPECT. Cine CMR image adapted by permission from Springer Nature from [287]. Coronary CT angiography image adapted by permission from Springer Nature from [67]. Myocardial perfusion SPECT image adapted by permission from Springer Nature from [109].

Echocardiography

Echocardiography is the cornerstone of imaging in cardiology. It is fast, relatively cheap and can be performed at the bedside, although most scans are prospectively planned and performed in dedicated echocardiography departments.

Echocardiography was first developed in 1955. Using a time-motion display of the ultrasound wave along a single line of the ultrasound beam, called M-mode imaging, it allowed a simple visualization of the contractile motion of the myocardium. Echocardiography has since developed significantly. Nowadays, a typical exam includes 2-D and even 3-D cine imaging of the heart chambers, interrogation of blood flow using pulsed or continuous wave Doppler signals, and myocardial wall motion velocities and strain using tissue Doppler and speckle tracking technology. Figure 2.7a shows a typical 2-D echocardiography scan with blood flow velocity measured using Doppler imaging shown as a colour overlay. The duration of a typical echo exam is approximately 15-20 minutes.

Due to its speed, mobility and costs, echocardiography is often the first imaging technique used to investigate cardiac function in patients with (suspected) heart disease. It allows a screening of ventricular size, assessment of contractile (systolic) and relaxation (diastolic) function, and interrogation of the anatomy and function of the heart valves. Except for 3-D imaging, most images are reconstructed from the sound wave reflections in real time, resulting in a sharp contrast between blood (black) and tissue (grey-white) at frame rates ranging from 40-120 frames per second depending, for example, on the width, depth and sample line density of the ultrasound beam. This fast imaging with sharp blood-tissue contrast and the presence of consistent speckle patterns makes echocardiography suitable for assessing fast cardiac events, and in particular the motion of the myocardium and heart valves, as well as flow acceleration and regurgitation through the diseased heart valves or stenotic regions of blood vessels.

Several limitations of echocardiography may strongly impact the use of machine learning techniques on these image sequences. Firstly, the ultrasound beam is hindered by the bony structures of the chest wall and air in the lungs. As a result, imaging planes are limited and the quality of the images can vary significantly between patients. This impedes accurate, reproducible measurements of cavity volumes to calculate EF, as well as impeding assessment of certain structures, notoriously atria and the right ventricle (RV),

from trans-thoracic echocardiography. Trans-oesophageal echocardiography reduces some of these disadvantages but is invasive. The second disadvantage is that echocardiography does not allow characterization of myocardial tissue structure and also provides no information about myocardial perfusion, which is an important factor in coronary artery disease, the most common disease in cardiology.

Despite these limitations, machine learning has started to be applied to the analysis of echocardiography images [25]. For example, machine learning models have been developed for automatically classifying standard view planes [251], quantification of cardiac function [134] and disease detection [435].

Cardiac MR

Cardiac magnetic resonance (CMR) is a more recently developed technique for imaging of the heart. In CMR, the spin speed and direction of hydrogen atoms is manipulated using magnetic gradients. Echoes of changes in electromagnetic charge are received by the scanner and utilized to construct images of the anatomical structures. In comparison with echocardiography, CMR allows imaging of the heart and all other structures in the chest, without being restricted by imaging windows or depth of the imaging beam. As a result, it allows for more reliable quantification of cardiac volumes and function. Moreover, as the signals are based on the quantity of hydrogen molecules in tissues, it also allows for characterization of the composition of the myocardium. This way, it can be used to detect fibrotic tissue (scars of previous ischemic events) or the presence of inflammation or molecular deposits in the tissue.

A typical CMR exam currently takes about 30-40 minutes. Multiple different image sequences are acquired to obtain all relevant information: cine imaging is used to acquire dynamic cardiac images and myocardial motion information, late gadolinium enhancement (LGE) imaging is used for scar detection and T1 and T2 maps are used for characterization of deposits and inflammation. CMR images are typically reconstructed using information obtained over multiple heartbeats. Therefore, breath-holds or breathing navigators are needed to ensure a similar position of the heart during acquisition. A sample frame from a cine CMR acquisition is shown in Figure 2.7b.

The main disadvantage of CMR is that MR machines are bulky and expen-

sive. Moreover, metal implants, such as internal defibrillators or pacemakers, cause distortions to the images and the narrow bore of the machine is challenging for patients experiencing claustrophobia.

In clinical practice, CMR exams are not currently used in the initial screening for heart diseases in patients. They are typically requested in patients with established heart disease in whom investigation of the underlying cause (using tissue characterization and scar detection) or reliable quantification of right and left ventricular volumes is needed to inform further treatment decisions.

The role of CMR in cardiology is still growing. The ever faster and higher quality of CMR images, as well as increased presence in clinical guidelines, is resulting in more patients being referred for CMR to investigate causes of heart failure or monitor treatments.

Because of the generally better image quality of CMR compared to echocardiography and the availability of large annotated databases, machine learning models for CMR analysis are more mature [226]. For example, robust models have been proposed for image reconstruction [152], segmentation [76] and automatic biomarker estimation with quality control [335]. Of relevance to such models is the fact that CMR images are typically acquired "slice-by-slice", and image resolution is normally good within-plane, but is less good through-plane. This has consequences for subsequent algorithms for image analysis, introducing extra uncertainty into measurements made in the through-plane direction. Furthermore, the 3-D nature of many CMR images also introduces extra computational cost if fully 3-D processing is attempted, and so many models instead limit themselves to 2-D slice-by-slice analysis. Processing 2-D slices also offers more images to train machine learning algorithms, but may raise spatial consistency issues that are currently under active research.

Cardiac CT

Computed tomography (CT) imaging utilizes X-ray radiation to create an image of the internal organs of the body. In cardiology, it is often used for static imaging of the structural anatomy of the heart and the structures related to it, such as the coronary arteries and great vessels. This use reflects the main benefit of CT: its high spatial resolution and good contrast between myocardium, blood and more calcified structures. The main application of cardiac CT is for qualitative and quantitative assessment of atherosclerotic

decompositions and stenosis in the coronary arteries. For example, a sample coronary angiograph is shown in Figure 2.7c. CT is also frequently used to investigate atrial anatomy prior to procedures that involve ablation or isolation of electrical foci of atrial fibrillation, or to assess structural abnormalities of the cardiac and vascular anatomy in patients with congenital heart disease. Dynamic imaging of the heart during contraction is possible using CT, but the significant radiation dosages involved currently make it less attractive than CMR. However, in patients with metal implants or claustrophobia, 4-D cardiac CT can be an option.

Machine learning models have been proposed for applications including reconstructing CT images from incomplete X-ray projection data [108], segmentation [76] and assessment of coronary artery disease [153].

Other Imaging Modalities

Single photon emission computed tomography (SPECT) and positron emission tomography (PET) are nuclear imaging techniques that can be used to quantify myocardial perfusion (see Figure 2.7d). Myocardial perfusion defects, originating from occlusive coronary artery or microvascular disease can be detected and quantified using these scans, similar to CMR perfusion imaging.

Radioisotopes are injected that emit gamma rays, which are detected by gamma cameras. By obtaining recordings at rest and during physical or pharmacological stress (which increases blood flow to the myocardium) myocardial perfusion defects can be detected. SPECT or PET are currently the standard option for myocardial perfusion assessment in many hospitals. However, the newer CMR perfusion exams have started to replace these techniques in some hospitals.

PET scans can also be used to investigate metabolic active tissues other than the myocardium, such as cardiac tumours or infections of endocardial structures (endocarditis).

Examples of the use of machine learning in these modalities include PET reconstruction [326] and prediction of coronary artery disease from SPECT [52].

Non-imaging Sources

Electrocardiogram

The electrocardiogram (ECG) is one of the earliest technologies developed to investigate the function of the heart. In 1903, Willem van Einthoven published his invention for use of the electrocardiogram and introduced the standard leads that allow investigation of the heart's electrical activity.

Myocytes are negatively charged with respect to their outside surroundings at rest. Contraction of the myocytes is activated by a rapid shift of electrolytes (most predominantly calcium ions) that results in depolarization of the cells. Consequently, relaxation of the heart is the result of myocyte repolarization due to a rapid reserve shift of calcium ions. The sum of changes in myocyte polarization in the heart can be detected using ECG. Moreover, the sequential activation of the cardiac structures (sinus node – atria – atrioventricular node – ventricles) results in a change in size and direction of the electrical field, and can be identified from the ECG traces. ECG signals are affected by size of the cardiac structures, myocardial muscle mass, muscle oxygenation and the speed of the activation wave front through the ventricles. An ECG is obtained using a small, mobile and cheap device and can be performed within a minute.

Due to its sensitivity for changes in cardiac structure or function and its ease of use, ECG recordings are one of the most used tests in cardiology. The ECG is the main diagnostic tool to identify arrhythmias (disturbances in the sequence of activation in the heart) and diagnose acute coronary artery disease (acute hypoxia and necrosis of myocytes). For patients with chronic cardiac disease ECG recordings are used to monitor changes in electrical activation that suggest disease progression.

Recently, some papers on machine learning based analysis of ECGs have started to emerge. One notable example is [323], who demonstrated how a convolutional neural network could predict 1-year all-cause mortality from 12-lead ECG signals. Other notable examples come from the *PhysioNet* and *Computing in Cardiology* communities, who organize public data challenges on ECG processing and diagnosis on a yearly basis. The 2020 challenge was extremely popular and involved more than 200 teams using machine learning algorithms to diagnose 12-lead ECG signals from several large databases totalling 66,000+ recordings [22].

Machine learning also offers relevant solutions for the modeling and analysis of electrophysiological data, including 3-D mappings acquired from catheter recordings [63] and personalized computational cardiac simulations [242]. These applications are discussed further in Chapter 10.

Electronic health records

In EHRs, doctors record all patient-related information in a systematic fashion. A typical daily report of a patient includes a medical history, details verbally given by the patient about his or her complaints, findings found during physical examination, a brief description of test results (e.g. important biochemical abnormalities or imaging findings), a conclusion and a treatment plan. Cardiologists use these detailed reports, made during every visit for an outpatient clinic or daily during in-hospital stays, to evidence their care, hand over between different professionals in the medical team and evaluate and register treatment effects. Apart from the reports written by doctors, the EHR also contains separate modules that display biochemistry lab results, ECG recordings, imaging exam result reports (such as a report of the analysed echo exam or CMR scan) and structured lists of contact moments (such as outpatient visits or admissions), previous diagnosis and current and previously prescribed medication. PACS systems are similar to the EHR, except that these are dedicated solutions for archiving of the acquired medical images and do not contain other information apart from those relevant to the images, such as patient identifiers.

The use of machine learning with EHRs has focused on two different applications: (i) automated generation of EHRs from imaging data [263], and (ii) machine learning based analysis of EHR data [270]. We review each of these fields in more detail in Chapter 10.

Closing Remarks

We hope that this chapter has provided the reader with a grounding in the fundamental concepts of traditional machine learning models, as well as an awareness of some of the potential pitfalls and difficulties developers might face and the data sources that such models typically exploit in cardiology. Next, we provide several exercises to let you self-test and reinforce your knowledge, followed by our first hands-on tutorial that we hope will help you to get started in your explorations of machine learning model develop-

ment.

However, as we saw in Chapter 1, much of the recent success of, and interest in, machine learning comes not from the types of traditional model that we have discussed in this chapter, but rather from models based upon artificial neural networks, or *deep learning*. In the next chapter, we introduce the fundamental theory behind such models, and also provide a tutorial to help you to develop your own neural network model.

Exercises

Exercise 1.
In what situations might an unsupervised machine learning model be an appropriate choice?

Exercise 2.
What imaging and non-imaging data are typically popular for the development of machine learning algorithms in cardiology? Are some more challenging than others and why?

Exercise 3.
A machine learning model has been developed for automated diagnosis of some types of cardiovascular disease based on CT images. To train the model, the developers have used a training set of 100 CT images and associated diagnoses. They have implemented a number of different supervised machine learning models, each with different hyperparameter settings. The best-performing model on a test set of 50 CT images and diagnoses has been chosen for deployment.

What concerns do you have about the validation strategy adopted by the developers? Would you expect the chosen model to perform as well when deployed on real clinical data?

Exercise 4.
A company is developing an automated tool to segment the aorta from CMR images, with a view to using the segmentations to derive functional biomarkers. The company plans to train a supervised segmentation model using annotations produced by manual contouring. However, the manual contouring process is very laborious and time-consuming. What alternative approach would you recommend?

Exercise 5.

An implantable cardioverter-defibrillator (ICD) is a small battery-powered device that is implanted in the chest to monitor heart rhythm and detect irregular heartbeats. An ICD can deliver electric shocks via one or more wires connected to the heart to fix abnormal heart rhythms. A research team is investigating more targeted use of ICDs to avoid unnecessary interventions. They would like to use machine learning to exploit routine clinical data in order to more accurately predict which patients are likely to suffer life-threatening arrythmias in the future.

Explain how you would go about designing a machine learning solution for this problem.

Exercise 6.
A clinical study is investigating whether automated measurements of global longitudinal left ventricular strain made from echocardiography can be a useful predictor of major adverse cardiac events (MACE - a composite endpoint that combines nonfatal stroke, nonfatal myocardial infarction and cardiovascular death). The team would like to use machine learning techniques in their study.

Suggest some ways in which machine learning could help in the study. What type(s) of model would be appropriate and how could they be validated?

Tutorial - Introduction to Python and Jupyter Notebooks

Tutorial 1.

As for the other notebooks, the contents of this notebook are accessible as Electronic Supplementary Material.

Overview

In this first hands-on tutorial, you will go through the basics of the Python language and objects. We will use a Jupyter Notebook, which is a very convenient didactic and interactive tool that can mix written explanations and sections of code. Our notebooks are tailored for a specific problem related to the chapter preceding each notebook. You will be asked to run existing sections of code, examine the outputs, and fill in missing code or adapt it to test different behaviours of an algorithm.

The figure below shows an example of an interactive Jupyter Notebook cell to be run in this notebook:

Instructions:

- **Observe in the code below how each `for` statement the code moves further to the left.**
- **Execute it with `Shift + Enter`: does the output correspond to what you expected?**

In [5]:

```
#the first code block
myfirstlist=[10,20,30,40]
mysecondlist=[1,2,3,4,5]

for tens in myfirstlist:
    #the second code block (a for loop over a list of integers)
    print('Entering second code block')
    for units in mysecondlist:
        #the third code block (a for loop over a second list of integers)
        new_number=tens+units
        print('Output of third code block; The new number is',new_number)

```

```
Entering second code block
Output of third code block; The new number is 11
Output of third code block; The new number is 12
Output of third code block; The new number is 13
```

Objectives

- *Become familiar with the basics of Python and Jupyter Notebooks.*
- *Understand the main objects (variables, functions, operators, etc.) that will be handled in the subsequent hands-on tutorials.*
- *Gain practice on simple illustrative exercises.*

Computing Requirements

Each notebook starts with a brief "System setting" section, which imports the necessary packages, installs the potentially missing ones, and imports our own modules.

You will need Python installed on your computer and a software tool to run the notebooks (we recommend for example the free software JupyterLab (https://jupyter.org/). We assume that you have already installed very common packages such as Numpy, Matplotlib, and scikit-learn. In case you are missing these packages, or another one, we recommend you to run the following command (here illustrated for one of these packages):

```
pip install scikit-learn
```

We hope you'll enjoy these contents!

Acknowledgements

ND was supported by the French ANR (LABEX PRIMES of Univ. Lyon [ANR-11-LABX-0063] within the program "Investissements d'Avenir" [ANR-11-IDEX-0007], and the JCJC project "MIC-MAC" [ANR-19-CE45-0005]).

EPA was supported by the EPSRC (EP/R005516/1) and by core funding from the Wellcome/EPSRC Centre for Medical Engineering (WT 203148/Z/16/Z).

BR was supported by the NIHR Cardiovascular MedTech Co-operative award to the Guy's and St Thomas' NHS Foundation Trust and Wellcome/EPSRC Centre for Medical Engineering at Kings College London (WT 203148/Z/16/Z).

AK was supported by the EPSRC (EP/P001009/1), the Wellcome/EPSRC Centre for Medical Engineering at the School of Biomedical Engineering and Imaging Sciences, King's College London (WT 203148/Z/16/Z) and the UKRI London Medical Imaging & Artificial Intelligence Centre for Value Based Healthcare.