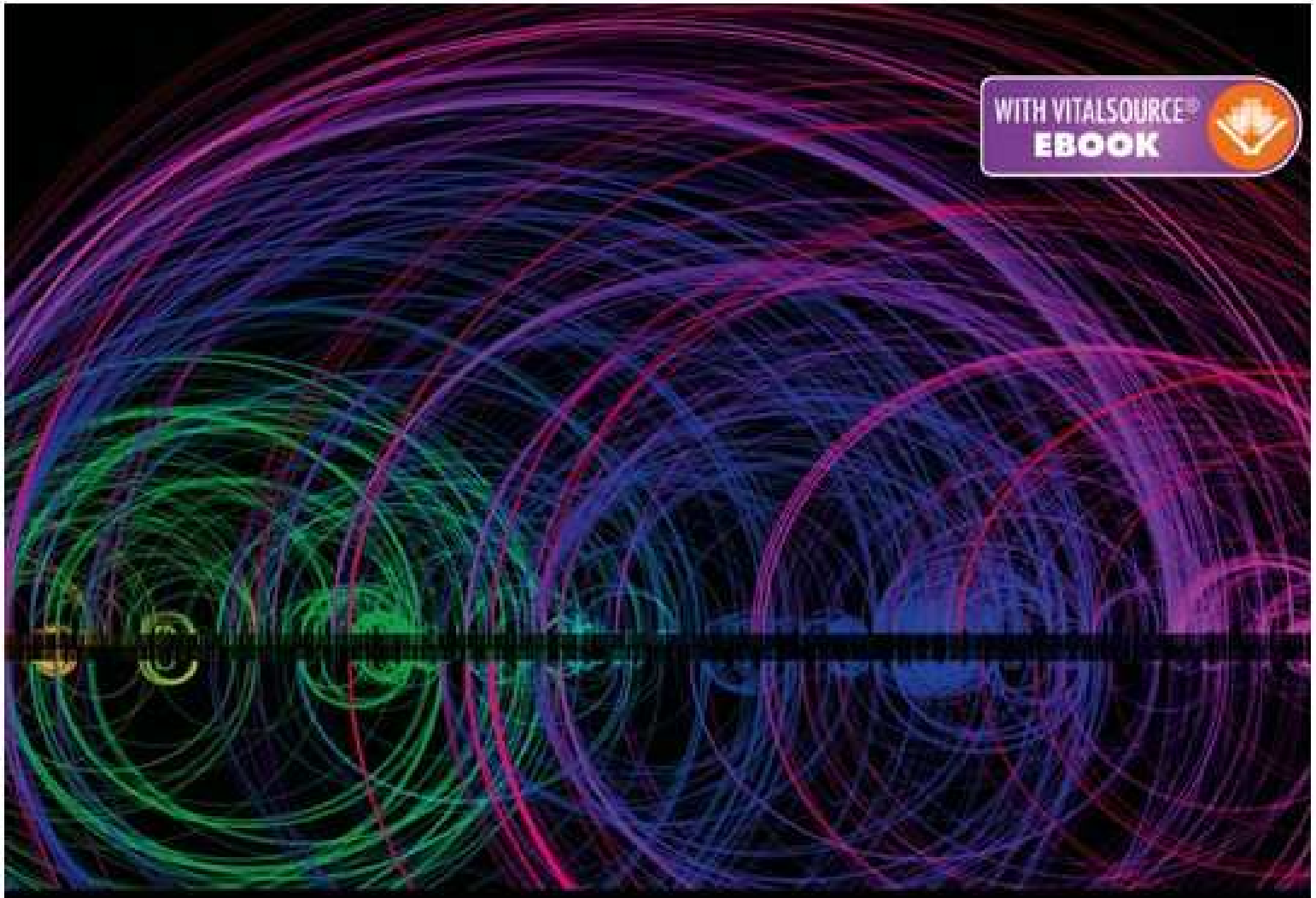

GUIDE TO HEALTH INFORMATICS

Third edition

Enrico Coiera

CRC Press
Taylor & Francis Group

CHAPTER 27 Model building for decision support, data analysis and scientific discovery

27.1 Before reasoning about the world, knowledge must be captured and represented

In Chapter 26, we were introduced to some of the different ways in which knowledge can be represented computationally and then used to draw automated inferences. In this chapter we explore the origins of that knowledge. In some situations, the knowledge embedded in a clinical decision support system (CDSS) is a direct translation of well-understood principles captured, for example, in guidelines or textbooks – the CDSS essentially replicates what is already known. In some circumstances, however, little may be known, or there is significant variation in system behaviour from one setting to another that has not been accounted for. In such circumstances, we use a different class of computer methods to assist in discovering new relationships in the world.

These *computational discovery* methods typically take as their input data from the process that needs to be understood, and they then output knowledge which may be in any of the forms we encountered in Chapter 26 – rules, networks, models. This encoded knowledge can then be placed into a CDSS capable of taking similar data as input and applying the knowledge to make a decision. Together, machine reasoning methods such as CDSSs and machine discovery systems complete the model cycle introduced in Chapter 1 from model construction through to use (Figure 27.1).

In this chapter, we review the different applications of machine discovery before exploring the main methods used to acquire knowledge, whether for use in a CDSS, to support the analysis of new data or to drive scientific discovery. The chapter also covers how one evaluates the correctness of these models and presents an approach to data analytics, based on machine discovery methods.

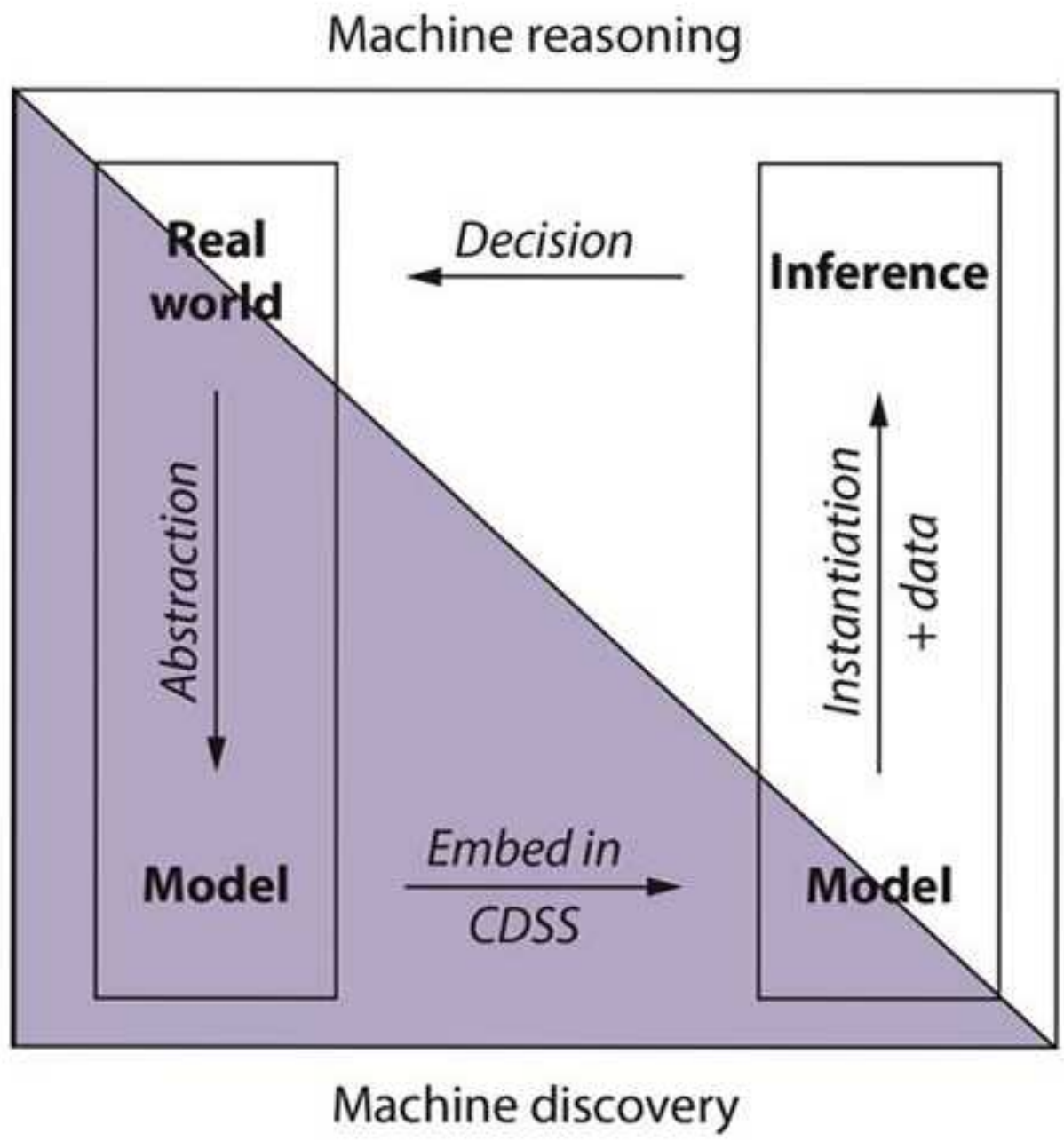

Figure 27.1 Computer reasoning depends first on acquiring models that record our understanding of how the world works and then applying those models in a decision support system to arrive at conclusions shaped by the data specific to a given decision task.

27.2 Computational discovery systems find wide application in healthcare

Computational discovery systems have three broad uses. The first is to create models that can then be used within a CDSS – a process sometimes called knowledge acquisition. Next, these systems can be used as an adjunct in the scientific discovery process, helping scientists understand the meaning of data. As data sets grow in size, machines start to become essential in identifying promising hypotheses to explain data patterns. The third application area is data analytics, in which we try to discover meaningful regularities in data, much as scientists do, but with the goal of using this information directly to help in managing a process such as running a hospital or health system. Where scientists engaging in discovery would treat data mining discoveries as hypotheses to be tested in new experiments, analysts would use their discoveries to intervene directly in the system they are trying to manage.

Association rule mining

Discovery systems can be used to develop the knowledge bases for a CDSS. Given a set of similar clinical cases that act as examples, a discovery system can try to identify which clinical features among the cases are most uniquely associated with a diagnosis, often expressed as rules. A classic example of this type of discovery system is KARDIO, which was developed to build and then interpret electrocardiograms (ECGs) (Bratko *et al.*, 1990). Association rules can be mined from historical electronic health records (EHRs), for example, to link medications with the clinical problems they most commonly treat. When new patients are admitted to hospital, the rules can use a patient's medication list to automatically generate the corresponding problem list (Wright *et al.*, 2013). Other discovery systems learn decision trees instead of rules (Quinlan, 1986). These trees look like the decision trees discussed in Chapter 8 and can be used to guide diagnosis or treatment decisions.

A similar approach can be used when *data mining* for previously unknown relationships hidden within large databases. For example, EHRs contain huge quantities of data, and buried with these records potentially lies much new knowledge (Jensen *et al.*, 2012). It may be that patients treated with a drug for one disease also gain benefit from the drug for another disease they have. This additional application of the drug may not be known, but by aggregating and comparing different patient groups, it should be possible to uncover the previously unexpected benefit. One can look for such associations in EHRs by searching for patients with a disease, looking within that group for patients who had a better prognosis than average, and seeing whether this group is associated with the prescription of medications for other diseases.

Previously unreported side effects of a drug can also be discovered by *text mining* adverse event reports or by looking for higher than expected rates of adverse events among patients taking a drug, compared with similar patients taking a different drug (Tatonetti *et al.*, 2011). Such text mining has application anywhere that we have assembled large databases of relatively unstructured text. One can look for patterns in the text of medical notes, including laboratory or imaging test reports. Adverse event reports, which are generated when patients are potentially or actually harmed during the process of care, are another rich source of information, and text mining can be used to discover important patterns in such data, as well as to generate CDSSs that automatically flag when a new report is serious and requires immediate attention (Ong *et al.*, 2012).

Process mining

It is often the case that there are several alternate treatments for a given condition, with slightly different outcomes. It may not be clear, however, which features of one particular treatment method are responsible for such variation in outcomes. Computational discovery methods have a role in the development of treatment guidelines, which essentially describe the most suitable treatment process for a patient (Abston *et al.*, 1997; Toussi *et al.*, 2009). *Process mining* is the general term for discovery techniques that take temporal sequences of events and use them to build models of common sequences. For example, extracting sequences of events captured in an EHR can be used to describe patient flows, such as the care pathway for patients who are admitted to a hospital and then move through a stroke unit or oncology ward (e.g. Huang *et al.*, 2013; Mans *et al.*, 2008; Mans *et al.*, 2009).

Process models may be represented by simple event sequences or more complex networks. When there is a pre-existing process description, such as a treatment guideline, then process mining methods can be used to look for how frequently the ideal process is executed and uncover common exceptions to the process. This information may suggest either the need to make local processes conform more closely to the guideline or to make changes to the guideline if the exceptions are well motivated (e.g. Mani *et al.*, 2007). Building process models can help explore the reasons behind variations in the execution of a specific procedure in the hands of different surgeons (Forestier *et al.*, 2012) or interventional radiographical studies such as angiography (Gentric *et al.*, 2013) and can contribute to skills assessment and targeted training.

455
456

Literature-based discovery

Discovery methods can be applied directly to the research literature, to uncover previously unsuspected associations. The Arrowsmith system is a classic literature-based discovery system, allowing a user to ask questions such as, 'Are there any things that are shared by patients with disease A as well as disease C?' Such questions allow insights about the management of one condition to be re-applied to the associated condition. In 1986, Don Swanson used Arrowsmith to connect papers about dietary fish oil with papers about Raynaud's syndrome. Some papers described how dietary fish oil reduced blood lipids, platelet aggregability, blood viscosity and vascular reactivity. Papers about Raynaud's syndrome noted that it was a peripheral circulatory disorder associated with and exacerbated by high platelet aggregability, high blood viscosity and vasoconstriction (Swanson, 1986). Swanson hypothesized that fish oil would thus be beneficial in the prevention and treatment of Raynaud's syndrome, a hypothesis that has since been tested and validated.

Biological model discovery

Beyond such associational discoveries, it is possible to use patient data to construct pathophysiological models that describe the functional relationships between various measurements. For example, a learning system can take real-time patient data obtained during cardiac bypass surgery and then create models of normal and abnormal cardiac physiology (Hau and Coiera, 1997). These models may be used to look for changes in a patient's condition if they are applied at the time they are created. Alternatively, if used in a research setting, these models can serve as initial hypotheses that can drive further experimentation. Model discovery finds great use in hypothesizing new genetic regulatory mechanisms, metabolic pathways and disease processes.

Drug discovery

One particularly exciting application of learning systems is to discover new drugs. The learning system is given examples of one or more drugs that weakly exhibit a particular activity, and based upon a description of the chemical structure of those compounds, the learning system suggests which of the chemical attributes are necessary for that pharmacological activity. Based upon the new characterization of chemical structure produced by the learning system, drug designers can try to design a new compound that has these characteristics. Traditionally, drug designers synthesized a number of analogues of the drug they wished to improve upon and experimented with these analogues to determine which exhibited the desired activity. By using discovery tools, the development of new drugs can be speeded up and the costs significantly reduced. Statistical analyses of drug activity have been used to assist with drug analogue development (exploring different molecular variations of a drug), and computational discovery techniques have been shown at least to equal if not outperform chemists, as well as having the benefit of generating knowledge in a form that is easily understood by them (King *et al.*, 1992).

456
457

27.3 Manual knowledge acquisition methods can guide interactions with human experts to reveal the basis for their judgements

The first CDSSs used manually developed knowledge bases, typically obtained by describing the reasoning used in the heuristic judgements of experts. These rules described the features associated with different diagnoses and were used to generate a differential diagnoses or suggest the most appropriate therapy for a clinical case. The methods used to craft these rules ranged from the highly informal to the systematic use of robust processes designed to elicit the rationale underpinning human judgements. Despite their often qualitative nature, these 'manual' methods remain an important model building tool, both used alone and to help guide automated processes. In the latter situation, although we may have huge data sets with very many data types, the process of automated model building often needs direction from those who understand the problem domain – for example, identifying which patterns are interesting or which data types are most likely to be significant in shaping an outcome.

There are several classic approaches to *knowledge acquisition* or *knowledge engineering*, and these are very similar to qualitative methods widely used by researchers when they seek to answer a variety of other questions (Cooke, 1999; Olson and Rueter, 1987; Welbank, 1990):

Interviews

Especially early in the process of understanding a decision task, interviews can be very illuminating. Both unstructured (free flowing) and structured (following a question script or describing a typical scenario) interview methods can be used. Structuring interviews is more appropriate when there are clear questions to be answered, and unstructured approaches are more useful in the very early sense-making stages of the process. Interview data can be collected on anything from note form through to full video recording.

Observation

By observing humans as they make judgements, either in real world settings or, for difficult situations, in response to artificial cases, we can develop an understanding of the types of decision tasks they face and how they resolve them. Observers can take on active, non-participatory or highly passive roles and capture what they observe through notes, audio, images or video. Many of the methods used to analyze data are borrowed from anthropology and ethnography. These include *grounded theory* (Strauss and Corbin, 1994), in which researchers gather data in as neutral a way as possible and then seek the concepts implied by the data by constant comparison across data types, supported by additional data gathering. Researchers take note of how frequently new concepts emerge from the data analysis and seek the point of *saturation* (when no further new concepts are evoked) to indicate that the knowledge acquisition task is likely complete. Advantages of observation are the richness of the data that can be captured and their reflection of reality, as opposed to recollection. Disadvantages include the difficulty in capturing rare events or understanding complex processes by observation alone.

457
458

Process tracing

Procedural descriptions can be obtained by observing a sequence of behaviours required to carry out a particular task. *Think-aloud protocols*, for example, ask subjects a to explain what they are thinking as they carry out a predetermined task, such as operate a device, use a piece of software or engage in a clinical process. In addition to capturing what is said and logging event sequences, subjects can also have a variety of other measurements taken, including eye tracking to see where they focus at any particular moment and physiological stress measurements if the setting is one in which operator stress is an issue. Some tasks may be so onerous that it is not possible to explain what is happening from moment to moment, and in these cases subjects are asked to recount what was happening later, perhaps in response to recorded data such as video. Process tracing methods generate potentially very rich data sets that can require substantial effort to analyze. Computational tools can assist in process analysis, for example, by looking at process traces from a number of different subjects and across different example cases and then generating typical pathways based upon statistical analyses. Such methods also find use in designing user interfaces for software because they reveal the paths that users take as they try to operate software, thus informing designers of the difficulties users face when confronted with their systems.

Conceptual methods

In order to understand a decision task completely, it is necessary to describe all the underlying concepts in the domain and their relationships. If we were eliciting knowledge about the diagnosis of a disease group, then the concepts would include the diseases that are under consideration, the clinical symptoms and signs that a clinician may uncover and the different tests available and their typical results for different diseases. Conceptual methods are formal approaches to capturing these underlying structures, and they tease apart the ontology and conceptual basis of a domain, as well as the knowledge about these relationships. Methods can assist in inducing subjects to explain whether specific concepts are related and to what degree. The *repertory grid* method asks subjects to rate the relatedness of concepts along highly structured dimensions. For example, when attempting to relate a clinical sign to a different disease, clinicians may be asked to make that judgement according to expected frequency, ease or cost of elicitation and whether the sign is pathognomonic.

Case-driven knowledge acquisition

Knowledge bases typically need to be updated as knowledge changes or as the role of a CDSS expands. As new rules or other elements are added, they can introduce inconsistencies, with new elements overlapping or even contradicting each other, a result in part because 'many hands' are involved. To circumvent this knowledge maintenance problem, computational tools can carefully guide the process of adding knowledge to a system. In the Ripple-Down Rules (RDR) methodology, experts start with an essentially empty CDSS and add new rules for each new case encountered that the CDSS cannot handle (Compton and Jansen, 1990). The new knowledge that 'repairs' the system's performance on a case is attached to the portion of the classification tree that failed. To ensure that the changes do not introduce errors, a set of cornerstone cases is run against the updated CDSS, containing the data describing each case and the expected 'gold standard' response to the case. Despite a process that is driven by new cases, rather than by abstract principles of form, and that seems to produce quite large and messy knowledge bases, CDSSs built using RDR appear much easier to build and maintain and to perform just as well (Compton *et al.*, 1992). Many commercial systems are based on RDR, for example, to assist with bulk interpretation of high-volume laboratory results (Compton *et al.*, 2006). Although initially developed to manage single classification tasks in pathology, RDR methods have evolved to support multiple classification tasks and have found application across a wide range of domains and problem types including configuration, simulation, planning and natural language processing (Richards, 2009b).

Crowdsourcing

Instead of relying on individual interviews or observations and leaving knowledge synthesis to the *knowledge engineer*, users and experts can engage collaboratively in the development of a knowledge base. Crowdsourcing can be used with great accuracy to identify key concepts and relationships between concepts in a domain, such as the different drugs typically prescribed for given clinical problems (McCoy *et al.*, 2012). Clinicians can also be asked to agree on their recommendations for the best decision in different cases, and these decisions then can be used as a gold standard against which to test a CDSS (Wagholikar *et al.*, 2013). Wiki-style collaborative tools have also been developed to support case-based RDR knowledge acquisition (Richards, 2009a). The value of the crowdsourced approach is that many different individuals can, in a controlled way, suggest key concepts and knowledge elements. These can then be communally critiqued, and new evidence can be brought in to correct misconceptions, all in a way that a generic knowledge engineer, unfamiliar with a specific domain, is unable to do.

Systematic review

All these methods presume that 'experts' working in a domain are the main sources of knowledge that is to be codified. It is often the case, however, that the decision task for a CDSS is well described in the scientific literature. As we saw at the end of Chapter 7, a formal process such as systematic review can be used to search for and select research studies into the efficacy of different treatments for a disease and to then synthesize the evidence so that a recommendation can be made about the best course of treatment in different circumstances. The resulting clinical guidelines are documents can then be translated in a computer representation such as rules or a protocol.

27.4 Computational discovery systems generate new knowledge from data

All scientists are familiar with the standard approach to data analysis. Given a particular hypothesis, statistical tests are carried out to see whether the hypothesized relationship can be found among specified variables in a data set. *Computational discovery systems* typically hypothesize such relationships among different variables on their own and then test to see whether these fit a data set. The methods for testing relationships include classic statistical methods, information theory and *machine learning*.

Supervised and unsupervised learning

The learning task for a machine can take two major forms. In many cases, a data set will come with some class labels attached to the data. For example, a set of patient records (the data) is categorized as normal or diabetic (the class labels). The discovery system then sets about trying to understand which attributes of the data set are uniquely associated with each class. In contrast, when data are unlabelled, the learning task is more complex because the machine needs to hypothesize its own classes.

Supervised learning algorithms are designed to learn from labelled examples. We provide the algorithm with examples of two or more classes (or labels) and ask it to find a model that best distinguishes between the classes. For example, a supervised algorithm can be tasked to distinguish patients who have a low risk of breast cancer from those with the disease, by using their medical records. A set of *positive examples* – patients with breast cancer – is identified manually from the medical record, typically by experts, along with a set of *negative examples* – patients who do not have the disease and appear at low risk. Each example is a data record built out of different data elements (e.g. blood pressure, age, biomarkers, expressed genes in a tissue), and these individual variables are called *features*.

The learning algorithm is then shown a portion of the data (including an equal number of positive and negative examples) as its *training set*. Using these data, the algorithm builds a model designed to best separate the classes, based upon the values of the features. For example, the algorithm might identify that possessing the gene *BRCA1* seems to be common to patients with breast cancer, but not the negative examples. To validate the model built by the algorithm, the remaining unseen portion of the data is used as a *test set*. The model developed in the training phase is applied to the data as if it was a functioning CDSS, and the machine-predicted class is compared with the true label. Once classification performance is acceptable, the learned model can be used in a CDSS.

In contrast, *unsupervised learning* algorithms have access only to an unlabelled training set. The purpose of these algorithms is to 'data mine', in which they try to discover new classes within the data, as well as to provide the descriptions of each class. For example, an unsupervised algorithm may look at the EHR data from patients with breast cancer and try to discover sub-groups of these patients with different disease presentations or treatment responses. The algorithm may, for example, discover that there is a group of patients with *BRCA1* and *BRCA2* genes whose prognosis and response to treatment are clearly different from those of other patients with breast cancer. The algorithm will try to identify the features that distinguish the sub-groups, and this information could then guide scientists to examine what biological or other explanation exists for this previously unseen difference.

When class labels are clear or the cost of obtaining the labels is justifiable, then a supervised learning approach makes sense. When labelling is difficult or impossible, unsupervised methods are the only possible approach.