Chapter 2
Mathematical Foundations of Decision Support Systems

S. Andrew Spooner

Abstract Health information technology can support decisions in a variety of ways, ranging from the passive display of information to intensive computation designed to model complex clinical reasoning. This chapter reviews the basics of the mathematics behind the methods that involve computation, including set theory, probability, Boolean logic, Bayesian reasoning, and nonknowledge-based systems.

Keywords Bayes theorem • Mathematics • Logic • Probability • Set theory • Electronic health records

Many computer applications may be considered to be clinical decision support systems. Programs that perform PubMed [1] searches do support decisions, but they are not "clinical decision support systems" in the usual sense. What we usually mean by a clinical decision support system (CDSS) is a program that supports a reasoning task carried out behind the scenes and based on clinical data. For example, a program that accepts thyroid panel results and generates a list of possible diagnoses is what we usually recognize as a *diagnostic* decision support system, a particular type of CDSS. General-purpose programs that accept clinical findings and generate diagnoses are typical diagnostic decision support systems. These programs employ numerical and logical techniques to convert clinical input into the kind of information that a physician might use in performing a diagnostic reasoning task. While one might suspect that such functionality might be useful within an electronic health record (EHR) system, this type of support is seldom found there; the usefulness of the EHR in decision support turns out to be less about sophisticated expert systems and more about access to needed information [2, 3].

Forms of decision support that are commonly found in EHR systems include alerts for medication prescribing [4], order sets that guide clinicians to use the correct antibiotic [5] to diagnostic decision support that converts findings into a list of diagnoses worth considering [6–8]. While the latter is not in widespread use; simpler methods like order sets, documentation templates, and drug alerts are practi-

S.A. Spooner, M.D., M.S. (✉)
Cincinnati Children's Hospital Medical Center,
3333 Burnet Avenue, MLC 9009, Cincinnati, OH 45229, USA
e-mail: andrew.spooner@cchmc.org

© Springer International Publishing Switzerland 2016

E.S. Berner (ed.), *Clinical Decision Support Systems*, Health Informatics,
DOI 10.1007/978-3-319-31913-1_2

cally universal in EHRs today. Nonetheless, the mathematics of diagnostic decision support is worth reviewing. Essential to the understanding of CDSS is familiarity with the basic principles of logic and probability. A brief review of these areas, followed by a description of a general model of CDSS, as well as exceptions to the model, will help in understanding how some CDSS perform reasoning tasks. We end with a discussion of the mathematical challenges in the evaluation of simple alerts as they are commonly deployed in EHRs today.

2.1 Review of Logic and Probability

2.1.1 Set Theory

A brief review of basic concepts in set theory is helpful in understanding logic, probability, and many other branches of mathematics. A *set* is a collection of unique objects. For example, the major Jones criteria [9] for patients at low risk for rheumatic fever is a set:

$$\text{JONES}-\text{CRITERIA}-\text{MAJOR} = \begin{Bmatrix} \text{carditis, migratory polyarthritis,} \\ \text{erythema marginatum, chorea,} \\ \text{subcutaneous nodules} \end{Bmatrix}$$

Likewise, the minor criteria make a set:

$$\text{JONES}-\text{CRITERIA}-\text{MINOR} = \begin{Bmatrix} \text{fever, arthralgia, elevated acute phase reactants,} \\ \text{prolonged P}-\text{R interval on electrocardiogram} \end{Bmatrix}$$

To complete our description of the Jones criteria, we need a third set:

$$\text{GROUP}-\text{A}-\text{STREP}-\text{EVIDENCE} = \begin{Bmatrix} \text{positive culture, positive rapid antigen,} \\ \text{antibody rise or elevation} \end{Bmatrix}$$

To apply the Jones criteria, one compares the patient's findings with the items in the various sets above. A patient is highly likely to have rheumatic fever if there is evidence of group A streptococcal infection *and* the patient has *either* two major criteria *or* one major and two minor criteria.

Each *element* or *member* of the set is distinguishable from the others. A *subset* is any collection of elements of a known set. Using the first of the criteria above, a patient must have a subset of clinical findings containing at least two of the elements of JONES-CRITERIA-MAJOR to meet the Jones criteria for rheumatic fever. If a patient has the clinical findings:

$$\text{FINDINGS} = \{\text{migratory polyarthritis}, \text{chorea}, \text{subcutaneous nodules}\}$$

then we say that FINDINGS is a subset of JONES-CRITERIA-MAJOR, or, in set terminology:

$$\text{FINDINGS} \subseteq \text{JONES} - \text{CRITERIA} - \text{MAJOR}$$

The *cardinality* or *size* of a set is simply the number of elements in the set. For our two examples, the cardinalities (written by placing a vertical bar before and after the symbol for the set) are:

$$|\text{FINDINGS}| = 3$$
$$|\text{JONES} - \text{CRITERIA} - \text{MAJOR}| = 5$$

The basic set operations are *intersection* and *union*. The intersection of two sets is the set of elements the two sets have in common. For example, if there is a patient with the following set of clinical findings:

$$\text{CLINICAL} - \text{FINDINGS} = \left\{\begin{array}{l}\text{heart murmur},\\ \text{migratory polyarthritis},\\ \text{chorea, subcutaneous nodules},\\ \text{cough}\end{array}\right\}$$

then the intersection of this set and JONES-CRITERIA-MAJOR is written:

$$\text{CLINICAL} - \text{FINDINGS} \cap \text{JONES} - \text{CRITERIA} - \text{MAJOR}$$

It is easy to see that the intersection of these two sets is simply the set FINDINGS. The union of two sets is the set of all elements that belong to either set. Since, by definition, a set's elements must be distinguishable from one another, the set resulting from the union of our patient's findings and the Jones major criteria is written:

$$\text{CLINICAL} - \text{FINDINGS} \cup \text{JONES} - \text{CRITERIA} - \text{MAJOR}$$
$$= \left\{\begin{array}{l}\text{heart murmur, migratory polyarthritis, chorea, subcutaneous}\\ \text{nodules, cough, carditis, erythema marginatum, chorea}\end{array}\right\}$$

Anyone who has done a PubMed search in which two sets of literature citations are combined has performed these set operations; the AND function in PubMed is like set intersection, and the OR function is like set union.

Diagnostic criteria like the Jones criteria are good examples of how sets can be used to represent diagnostic rules. The full low-risk Jones criteria, represented in set theoretical terminology, might read like this (assuming we have sets JONES-

CRITERIA-MINOR and GROUP-A-STREP-EVIDENCE as described at the beginning of this section):

If CLINICAL-FINDINGS is the set of a given patient's symptoms, signs, and laboratory test results, then the patient is highly likely to have rheumatic fever if either of two conditions are met:

$$|\text{CLINICAL} - \text{FINDINGS} \cap \text{JONES} - \text{CRITERIA} - \text{MAJOR}| \geq 2 \quad (2.1)$$

and

$$|\text{CLINICAL} - \text{FINDINGS} \cap \text{GROUP} - \text{A} - \text{STREP} - \text{EVIDENCE}| \geq 1$$

$$|\text{CLINICAL} - \text{FINDINGS} \cap \text{JONES} - \text{CRITERIA} - \text{MAJOR}| = 1 \quad (2.2)$$

and

$$|\text{CLINICAL} - \text{FINDINGS} \cap \text{JONES} - \text{CRITERIA} - \text{MINOR}| \geq 2$$

and

$$|\text{CLINICAL} - \text{FINDINGS} \cap \text{GROUP} - \text{A} - \text{STREP} - \text{EVIDENCE}| \geq 1$$

There are other set operations besides union and intersection. For example, the phenomenon of *set covering* has applications in decision making [10]. A cover of a set is a set of subsets in which each element of the covered set appears at least once as a member of one of the sets in the cover set. An example makes this definition clearer. Suppose you were asked to recommend a list of antibiotics for your hospital's emergency department. Your objective is to stock the minimum number of antibiotics that will be effective for 95 % of the pathogenic organisms you've found in cultures at your hospital. For the sake of simplicity, suppose that there are six pathogens, each designated by a letter, which account for 95 % of the infections seen in your hospital.

You might represent this set of pathogens as:

$$\text{PATHOGENS} = \{\text{A, B, C, D, E, F}\}$$

You have the following set of antibiotics from which to choose:

$$\text{ANTIBIOTICS} = \{\text{A} - \text{Cillin}, \text{B} - \text{Cillin}, \text{C} - \text{Cillin}, \text{D} - \text{Cillin}, \text{E} - \text{Cillin}, \text{F} - \text{Cillin}\}$$

Each antibiotic is described by the set of pathogens for which that antibiotic is effective. Here is a list of your antibiotics, with their covered pathogen sets (each of which is a subset of PATHOGENS):

$$\begin{aligned}
&\text{A} - \text{Cillin} = \{\text{A}, \text{C}\}\\
&\text{B} - \text{Cillin} = \{\text{A}, \text{B}, \text{E}\}\\
&\text{C} - \text{Cillin} = \{\text{C}, \text{D}, \text{E}\}\\
&\text{D} - \text{Cillin} = \{\text{F}\}\\
&\text{E} - \text{Cillin} = \{\text{B}, \text{D}, \text{F}\}\\
&\text{F} - \text{Cillin} = \{\text{E}\}
\end{aligned}$$

What you seek is a set cover of the set PATHOGENS; in other words, you want to pick a set of antibiotics which contains at least one antibiotic that is effective for each pathogen. It's clear that all six antibiotics taken together make a set cover, but your job is to find the minimum number of antibiotics that will get the job done. Casual inspection shows that the set {A-Cillin, E-Cillin, F-Cillin} does the job as a set cover, in that at least one antibiotic in that set is effective for each one of the pathogens in PATHOGENS.

There are many other set operations which can be applied to real-world decision problems, but the brief introduction presented here should suffice to illuminate the concepts presented in this book. Generally speaking, sets are used to formalize logical operations in a way that a machine—usually a computer—can understand.

Before we leave the topic of sets, fuzzy sets are worth a brief mention. Under conventional principles of set theory, an element is either a member of a set or it isn't. Heart murmur, for example, is definitely not a member of the set JONES-CRITERIA-MAJOR. Under *fuzzy set theory*, membership in a set is not an all or nothing phenomenon. In a fuzzy set, an element is a member of the set with a certain probability; e.g., cough is a member of the set COLD-SYMPTOMS with a probability of 80 % (a four out of five chance). Fuzzy set theory has created new ways of looking at sets and new methods for applying set theory to solve decision-making problems: fuzzy logic [11–13]. Fuzzy logic has been used to tackle decision-making problems in which uncertainty plays a role.

2.1.2 Boolean Logic

Anyone who has performed a search of the medical literature using the PubMed system has used logic. When referring to common logical operations like combining two sets of literature citations using AND or OR, we often refer to these operations as "Boolean" logic, in honor of George Boole (1815–1864), a British mathematician who published seminal works on formal logic. Indeed, PubMed is not a bad way to learn about Boolean algebra, since its connection to set theory is made so clear by the sets of literature citations that we manipulate in that system.

Suppose we have performed two literature searches. The result of one search, set A, represents all the literature citations in the PubMed database that relate to rheumatoid arthritis. Set B consists of all the literature citations on immune globulin. By

asking the PubMed program to give us a new set that is the result of combining A and B using the AND operator, we have a new set, C, that contains literature citations on the use of immune globulin in rheumatoid arthritis. When we combine two sets of citations using the AND function of our PubMed program, we are asking the computer to give us all citations that appear in both sets. This corresponds roughly to the English use of the word *and*.

The word OR in Boolean logic has a slightly different meaning than in English. In everyday usage, *or* usually has an exclusive meaning; the statement, "You may opt for chemotherapy or radiation therapy," usually means that one may have one or the other therapy, but not both. The Boolean OR is different. If one were to perform another pair of PubMed searches, this time for all articles that have asthma as a keyword (set A) and those that mention "reactive airway disease" in the text of the abstract (set B), one could combine sets A and B with the OR function to get a comprehensive set of citations on asthma. Because the OR function takes all citations that appear in one or both of sets A and B, the OR function is said to be *inclusive*.

There are other Boolean operators, like XOR (exclusive OR: "either A or B but not both") and NAND ("A and not B"), but AND and OR are the basic operators with which we are familiar.

How is Boolean logic used in CDSS? The mathematical subjects of statement logic and predicate logic give us formal definitions of how statements can be combined to produce new conclusions. For example, consider the following statements:

1. Urine cultures with colony counts of 10,000 or more are considered positive if they are obtained by bladder catheterization.
2. This patient's urine culture shows more than 10,000 colonies of E. coli.
3. All patients with positive urine cultures should be treated for urinary tract infections.

The statements can be combined intuitively, without the use of formal mathematics, into the conclusion:

This patient needs to be treated for a UTI.

The logic that gave us the conclusion so easily, comes from our medical intuition, but computers have no intuition. They must be programmed to generate even the most obvious conclusions. To understand logic as it is implemented on a computer, one must understand the basics of predicate logic and deductive reasoning.

The above example about UTIs is a sloppy instance of a syllogism. A syllogism is a form of deductive reasoning consisting of a major premise, a minor premise, and a conclusion. The premises are combined, using rules of predicate logic, into a conclusion. For example, a syllogism in a ventilator management decision support system might be:

Major Premise: All blood gas determinations that show carbon dioxide to be abnormally low indicate an over-ventilated patient.
Minor Premise: The current patient's carbon dioxide is abnormally low.
Conclusion: Therefore, the current patient is over-ventilated.

Again, this conclusion is obvious, but by representing the above syllogism using symbols, where the symbol Low-CO_2 represents the state of abnormally low carbon dioxide and the symbol OVERVENTILATED represents the state of an over-ventilated patient, the syllogism looks more computer friendly:

$$\begin{aligned}&\text{Major Premise} : \text{Low} - \text{CO2} \Rightarrow \text{OVERVENTILATED}\\&\text{Minor Premise} : \text{Low} - \text{CO2}\\&\text{Conclusion} : \text{OVERVENTILATED}\end{aligned}$$

Extending this example, suppose we have another statement in our CDSS that over-ventilation should cause a High Rate alarm to sound (we can represent this by the symbol HIGH-RATE-ALARM), then we can construct the syllogism:

$$\begin{aligned}&\text{Major Premise} : \text{Low} - \text{CO2} \Rightarrow \text{OVERVENTILATED}\\&\text{Minor Premise} : \text{OVERVENTILATED} \Rightarrow \text{HIGH} - \text{RATE} - \text{ALARM}\\&\text{Conclusion} : \text{Low} - \text{CO2} \Rightarrow \text{HIGH} - \text{RATE} - \text{ALARM}\end{aligned}$$

Thus, we have generated a new rule for the system, where the intermediate state of overventilation is bypassed. This simplification of two rules into a new one may or may not help our understanding of the system, but the results the system gives are the same: a low carbon dioxide value sets off the High Rate alarm. One can imagine how large sets of rules can be combined with each other to reduce complex reasoning tasks to simple ones.

The syllogism above is an example of rule chaining, where two rules are chained together to form a new conclusion. Specifically, the simple system outlined above is a *forward-chaining deduction system*, because the system starts with *if* statements and moves to a *then* statement. In real life, though, we often start with the "then" portion of a logical rule. For instance, consider the clinical rule:

If your patient has asthma, then give an influenza immunization each fall.

There are many other rules in real clinical practice with the same "then" portion ("give a flu vaccine"). The question a clinician might ask is not "Does this patient have asthma? If so, I should give a flu shot," but more likely the question would be simply "Does this patient need a flu shot?" We start with the "then" portion of this set of flu shot rules. A *backward-chaining deduction system* does this—it starts with the "then" end of a set of rules and works backwards to answer questions based on its rule set. In the flu shot example, a backward-chaining system would start with the "Does this patient need a flu shot" question and immediately learn that the diagnosis of asthma would cause this rule to be satisfied. The system might then ask the user or query a clinical database about the presence of this diagnosis.

An example of a backward-chaining deduction system in medicine was the MYCIN system developed at Stanford [14], MYCIN's domain was the selection of antibiotics for the treatment of bacterial infections based on clinical and microbiological information. An example of a forward-chaining system in medicine was

GermWatcher, developed at Barnes Hospital in St. Louis, [15, 16] GermWatcher used as its rules the Centers for Disease Control and Prevention's National Nosocomial Infections Surveillance System [17]. Using a forward-chaining reasoning system called CLIPS (C Language Integrated Production System, Software Technology Branch, National Aeronautics and Space Administration, Johnson Space Center, Houston, TX), expert system shell GermWatcher worked in a large hospital microbiology laboratory to identify hospital-acquired infections early from culture data.

CDSS that use logic like the simple management system above have limited application, since the range of truth encompassed by this logical system includes only true (e.g., the High Rate alarm needs to be sounded) or false (e.g., the High Rate alarm does not need to be sounded). Not many applications in medicine can be reduced to such simple truths. There may be situations where the High Rate alarm might not always have to be sounded for a low carbon dioxide reading (e.g., for a head injury patient who needs low carbon dioxide to preserve cerebral blood flow). To accommodate these situations, it would be helpful if the response from the system were something like "the high rate alarm should probably be sounded." Such a system would then need to be able to handle probabilities, as well as certainties, which most CDSS do. MYCIN, for example, reported its conclusions in terms of their likelihood. The next section covers basic concepts of probability.

2.1.3 Probability

Everyday medical practice contains many examples of probability. We often use words such as *probably*, *unlikely*, *certainly*, or *almost certainly* in all conversations with patients. We only rarely attach numbers to these terms, but computerized systems must use some numerical representation of likelihood in order to combine statements into conclusions.

Probability is represented numerically by a number between 0 and 1. Statements with a probability of 0 are false. Statements with a probability of 1 are true. Most statements from real life fall somewhere in the middle. A probability of 0.5 or 50 % is just as likely to be true as false. A round, opacified area seen in the lungs on a chest radiograph is probably pneumonia; one might assign a probability of 0.8, or 80 %, (a four in five chance) to this statement. Based on the high probability of pneumonia, one might elect to treat this condition without performing further testing—a lung biopsy, perhaps—that would increase the probability of pneumonia to greater than 80 %. We are accustomed to accepting the fact that our diagnoses have a certain probability of being wrong, so we counsel patients about what to do in the event (we might use the term "unlikely event") that things don't work out in the expected way.

Probabilities can be combined to yield new probabilities. For example, the two statements:

$$\Pr(\text{diabetes}) = 0.6$$
$$\Pr(\text{hypertension}) = 0.3$$

mean that the probability of diabetes is 0.6, or 60 %, (three in five chance), and the probability of hypertension is 0.3, or 30 %, (three in ten chance). We have not specified the clinical context of these statements, but suppose these probabilities applied to a particular population. Suppose further that the two conditions are independent; that is, the likelihood of patients having one disease is unaffected by whether they have the other (not always a safe assumption!). If we then want to know what the probability is of finding a patient in our specified population with both diseases, we simply multiply the two probabilities (0.6 and 0.3) to get 0.18, or 18 %. If the two clinical conditions are not independent, (e.g., pulmonary emphysema and lung cancer) then we cannot combine the probabilities in such a simple, multiplicative manner. This is much like the AND function in PubMed or the interaction function as applied to sets.

The familiar "OR" function from our PubMed program also has a mathematical meaning in combining probabilities. If we wanted to know how many patients in the above example had diabetes *or* hypertension (remember: this would also include those with both diseases in the usual mathematical sense of *or*), we would compute:

$$\begin{aligned}\Pr(\text{diabetes OR hypertension}) = \Pr(\text{diabetes}) + \Pr(\text{hypertension}) \\ - \Pr(\text{diabetes AND hypertension})\end{aligned}$$

The last term in the above equation we already know to be $0.6 \times 0.3 = 0.18$, so:

$$\Pr(\text{diabetes OR hypertension}) = 0.6 + 0.3 - 0.18 = 0.72.$$

Conditional probability is another type of probability often used in medicine. A conditional probability is the probability of an event (or the probability of the truth of a statement) *given the occurrence of another event* (or the truth of another statement). The most familiar case of conditional probability in medicine arises in the interpretation of diagnostic tests. For example, the probability of pneumonia given a round density on a chest radiograph is what we need to know in interpreting that diagnostic test if it is positive. In mathematical notation, this conditional probability is written this way:

$$\Pr(\text{Pneumonia} \mid \text{Round Density on CXR}).$$

One reads this notation, "The probability of pneumonia given a round density on chest radiograph." This notation is convenient in the explanation of Bayes' rule, which is the cornerstone of the logic in many decision support systems.

2.1.4 Bayes' Rule

If we have a patient with jaundice, how likely is it that he has hepatitis? Written another way, we seek to learn:

$$\Pr(\text{hepatitis} \mid \text{jaundice}),$$

which is read as "the probability of hepatitis given the presence of jaundice." We may not have this probability at our fingertips, but we might be able to find a slightly different probability more easily:

$$\Pr(\text{jaundice} \mid \text{hepatitis}),$$

which is, simply, the probability of jaundice given the presence of hepatitis. The latter probability could be found by studying a series of patients with proven hepatitis (it would be easy to get this data by looking up diagnosis codes in the medical records department) and computing the percentage of these patients who present with jaundice. However, this does not directly answer our original question. Bayes' rule allows us to compute the probability we *really* want—Pr(hepatitis | jaundice)—with the help of the more readily available number Pr(jaundice | hepatitis). Bayes' rule [18] is simply this:

$$\Pr(\text{hepatitis} \mid \text{jaundice}) = \frac{\Pr(\text{hepatitis}) \times \Pr(\text{jaundice} \mid \text{hepatitis})}{\Pr(\text{jaundice})}$$

Notice that to solve this equation, we need not only Pr(jaundice | hepatitis), but Pr(hepatitis)—the probability of hepatitis independent of any given symptom—and Pr(jaundice)—the probability of jaundice independent of any particular disease. These two independent probabilities are called *prior probabilities*, since they are the probabilities prior to the consideration of other factors.

The derivation of Bayes' rule is very simple. We already know that the probability of any two events occurring simultaneously is simply the product of their individual probabilities. For example, the joint probability we already computed of diabetes and hypertension in a hypothetical population was:

$$\begin{aligned}&\Pr(\text{diabetes AND hypertension}) = \Pr(\text{diabetes}) \times \Pr(\text{hypertension})\\ &= 0.6 \times 0.3 = 0.18.\end{aligned}$$

We were free to multiply these together, because in our hypothetical population, the likelihood of one disease occurring in an individual was independent of the other. In other words:

$$\Pr(\text{hypertension}) = \Pr(\text{hypertension} \mid \text{diabetes}) \text{ and}$$
$$\Pr(\text{diabetes}) = \Pr(\text{diabetes} \mid \text{hypertension}).$$

In this population, one's chance of having one disease is unaffected by the presence of the other disease.

In medicine, we are often faced with the question of the likelihood of two interrelated events occurring simultaneously in a patient. The case of a diagnostic test and the disease it is supposed to test for is a good example: what is the probability of an abnormal chest radiograph and pneumonia occurring in the same patient simultaneously? The question asks for this probability:

$$\Pr(\text{pneumonia AND abnormal CXR}).$$

Can't we simply find out what the incidence of pneumonia in the population is, and multiply it by the incidence of abnormal chest radiographs in the population? A moment's reflection should show that this simple calculation is not sufficient. For example, if the incidence of pneumonia is 1 in 1,000, and the incidence of abnormal chest radiograph is 1 in 100, then the erroneous probability would be computed:

$$\text{WRONG: } \Pr(\text{pneumonia AND abnormal CXR}) = 1/1000 \times 1/100 =$$
$$0.00001 = 0.001\%$$

This does not fit with our clinical intuition very well, since we know that people with pneumonia tend to have abnormal chest films. Our intuition says that the probability of the two events occurring together should be pretty close to the probability of having pneumonia alone, since a majority of those patients will have abnormal chest films. What we *really* need to compute is this:

$$\Pr(\text{pneumonia AND abnormal CXR}) = \Pr(\text{pneumonia}) \times \Pr\begin{pmatrix}\text{abnormal CXR} \mid \\ \text{pneumonia}\end{pmatrix}.$$

This is the probability of pneumonia multiplied by the probability of an abnormal chest radiograph given that pneumonia exists. If we take Pr(abnormal CXR | pneumonia) to be 90 %, then the computation matches our intuition much better.

In general, for any two events A and B:

$$\Pr(\text{A AND B}) = \Pr(\text{A}) \times \Pr(\text{B} \mid \text{A}) \text{and}$$
$$\Pr(\text{B AND A}) = \Pr(\text{B}) \times \Pr(\text{A} \mid \text{B}).$$

But since Pr(A AND B) must surely equal Pr(B AND A), we can say that the right-hand sides of the equations above are equal to each other:

$$\Pr(\text{A}) \times \Pr(\text{B} \mid \text{A}) = \Pr(\text{B}) \times \Pr(\text{A} \mid \text{B})$$

Rearranging this equation, we have Bayes' rule:

$$\Pr(A \mid B) = \frac{\Pr(A) \times \Pr(B \mid A)}{\Pr(B)}$$

At an intuitive level, we use Bayes' rule when making seat-of-the-pants estimates of disease probability in patients. For example, if we designate hepatitis by A and jaundice by B, and there were an ongoing epidemic of hepatitis (i.e., Pr(A) was high), then our index of suspicion for hepatitis in a jaundiced person would be increased. Likewise, if the likelihood of jaundice due to other causes was high (i.e., Pr(B) was high), then our estimation of the probability of hepatitis as a specific diagnosis would be lowered. Similarly, if jaundice were pathognomonic of hepatitis (i.e., Pr(A | B) was 1 or near to it), then our hepatitis diagnosis would be greatly increased. By using numerical estimates of the probability of diseases, findings, and conditional probabilities, Bayes' rule can help make medical decisions.

One might imagine a simple CDSS in which one enters a single symptom and receives the probability of the presence of a disease given that symptom. A problem arises when one wishes to get disease probabilities given multiple symptoms. The number of data points needed to do Bayesian calculations on multiple simultaneous symptoms is huge. For example, in a system which handles only single symptoms, if one had a database of 1,000 symptoms and 200 diseases, one would need to create $1{,}000 \times 200 = 200{,}000$ conditional probabilities, 1,000 symptom probabilities, and 200 disease probabilities, for a total of about 200,000 numbers. Since most of these numbers are 0 (many symptoms are unrelated to many diseases), this may be a reasonable amount of numbers to collect into a knowledge base. When one starts considering the probabilities needed to do computations on two simultaneous symptoms, this number climbs from 200,000 to about 200,000,000! If one wanted to design a system that could handle the very realistic situation of five or six simultaneous symptoms, estimating the number of numbers needed to support the calculation would be intractable. Modifying the system to handle multiple simultaneous "diseases" adds even more to the complexity. Only after making the simplifying assumption that most disease findings are independent of one another [19] do many diagnostic CDSS use Bayesian approaches. One such system, Iliad [20], successfully employed this assumption.

2.1.5 Informal Logic

Even if we create a reasoning system that follows all the rules of logic and probability, it would be difficult to come up with all the numbers that must be assigned to each event in even a small clinical database. Many successful CDSS have circumvented this difficulty by employing informal rules of logic to accomplish the reasoning task, without creating an intractable data gathering task. In the early development

of one of the most famous CDSS, MYCIN [14, 21, 22], the creators of the system developed their own logic system (heuristic) that made intuitive sense. This system employed "certainty factors" which ranged from −1 (false) to +1 (true). A certainty factor of 0 indicated no belief in either direction in the statement's veracity. In combining several statements with the AND function into a single combined statement in MYCIN, one simply takes the minimum certainty factor of all the statements as the certainty factor of the combined statement. This makes a certain intuitive sense: we cannot be any more certain of an AND statement than we are of the least certain part. Later development of the MYCIN project showed a sound probabilistic basis for the certainty factor rules, but the point here is that sometimes cutting mathematical corners can still yield a useful system. In other early CDSS (QMR [23] and DXplain, [8, 24]), there is a knowledge base of diseases and findings (a finding is an item from the history, physical examination, laboratory data, or radiographic data). Each disease is defined by a particular set of findings. Each disease-finding relationship is assigned a frequency (of the finding among people with the disease) and an evoking strength (of how strongly a finding would evoke the possibility of a disease) on an ordinal scale (1–5 for frequency; 0–5 for evoking strength). These two factors make intuitive sense, and the system works, but the manipulation of these factors within these systems is very different from the formal algebra of logic and probability.

2.2 The General Model of Knowledge-Based Decision Support Systems

There are similarities between physician and CDSS reasoning, although a CDSS might arrive at a similar conclusion to a physician without employing the same model of reasoning. Physicians do use some probabilistic information when they make decisions. For instance, a physician might make a diagnosis of influenza more often during the winter when influenza is more prevalent (probable) than in the summer. However, physicians use this information in informal ways; in other words, they do not use precise numbers in formulas to make diagnostic decisions [25, 26]. Another feature of real-life clinical decision making is that physicians do not require complete information to make a decision. Most doctors are comfortable making decisions based on incomplete or contradictory information [27]. In contrast, CDSS rely on well defined numerical techniques to do their reasoning, and they do require sufficient information to complete their formulae. While physicians can fall back on their knowledge of pathophysiology, CDSS are not well suited to situations in which hard data are unknown. To understand how these systems operate, and under what conditions they are best used, it is important to appreciate a general model of CDSS.

Figure 2.1 shows a general model of a CDSS. There is input to the system and output from it. The CDSS has a reasoning (inference) engine and a knowledge base. Understanding these basic components provides a useful framework for under-

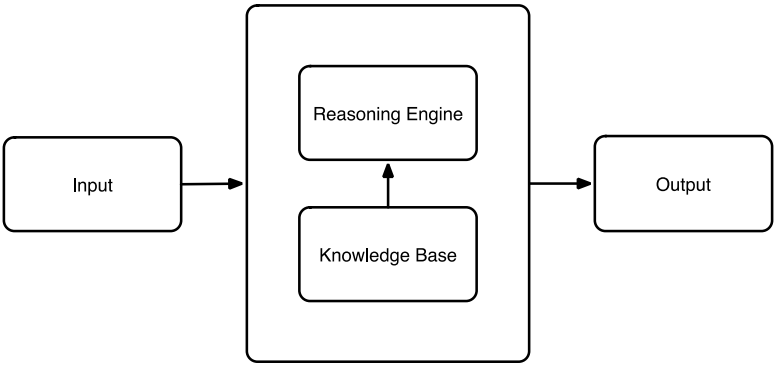

Fig. 2.1 A general model of a clinical diagnostic decision support system

standing most CDSS and their limitations. There are systems which do not follow this model which will be discussed briefly later in this chapter and in Chap. 3 in more detail.

The user supplies input appropriate to the system (i.e., terms from the system's controlled vocabulary to represent clinical data), and the system supplies output (e.g., a differential diagnosis or a therapy suggestion). The reasoning engine applies formal or informal rules of logic to the input and often relies on additional facts encoded in the system's knowledge base. The knowledge base is the compilation of the relationships between all of the diseases in the system and their associated manifestations (e.g., signs, symptoms, laboratory and radiographic tests). Maintaining the knowledge base is the most significant bottleneck in the maintenance of such systems, since the knowledge base needs to be expanded and updated as medical knowledge grows.

2.2.1 Input

The manner in which clinical information is entered into the CDSS (user interface) varies from system to system, but most diagnostic systems require the user to select terms from their specialized, controlled vocabulary. Comprehension of natural language has been an elusive goal in the development of CDSS. While it would be highly desirable to be able to speak or type the query "What are the diagnostic possibilities for a 4-year-old child with joint swelling and fever for a month," most who have used such systems are accustomed to the task of reformatting this question in terms the particular CDSS can understand. We might, for example, break the above query into components:

- Age: 4 years
- Gender: unspecified
- Symptom: joint swelling

- Duration: 1 month
- Time course: unknown

This breakdown of the original query might work on one system, but another system might demand that we break it down another way:

- Age: less than 12 years
- Finding: arthritis

Notice that the second description describes the age in vague terms, and it forces us to eschew joint swelling for the more specific term arthritis (usually defined as joint pain, redness, warmth, and swelling). In the vocabulary of the program, the age of 4 years (as opposed to 10 years) is unimportant, and joint swelling, without other signs of inflammation, is undefined.

Any physician who has assigned diagnostic and procedural codes in billing systems understands the limitations of controlled vocabularies. In a CDSS, it is common for the user's input to be restricted to a finite set of terms and modifiers. How well the system works in a given clinical situation may depend on how well the system's vocabulary matches the terms the clinician uses. CDSS take a variety of terms, called findings, which encompass items from the medical history, physical examination, laboratory results, and other pieces of clinical information. What constitutes a valid finding in a given program is entirely up to the program; there is no "standard" set of findings for all CDSS. For general purpose CDSS, items from the history and physical examination are going to be the findings. In specialized domains, e.g., an arterial–blood–gas expert system, the input vocabulary will be entirely different and much more restrictive.

Entering "chest pain" as a finding in a CDSS may be insufficient to capture the essence of the symptom. "Chest pain radiating to the left arm" may be sufficient, but usually there are pertinent temporal factors related to symptoms that are difficult to express in a controlled vocabulary. For example, "sudden onset, 20 min ago, of chest pain radiating to the left arm" has a very different meaning from "five-year history of continuous chest pain radiating to the left arm." While CDSS often include a vocabulary of severity and location modifiers, temporal modifiers are more difficult to build into a system, since minute changes in the timing of onset and duration can make a big difference in the conclusion the system reaches. Some CDSS make simplifying assumptions about broad categories of timing (acute, subacute, chronic) to aid in the temporal description of findings. Although users may experience frustration in being unable to enter temporal information, the research is equivocal in its impact.

One solution to the problem of temporal modeling in CDSS is to use an explicit model of time, in which the user is asked to specify intervals and points in time, along with temporal relationships between events (e.g., event A occurred before event B), in order to drive a temporal reasoning process within the CDSS. Clearly, this complicates the matter of entering data (to say nothing of programming the system). A simpler approach is to model time implicitly. In implicit time [28], temporal information is built into the data input elements of the CDSS; no special tem-

poral reasoning procedures are required. For example, one input item could be "history of recent exposure to strep." By joining the concept "history of" with the concept of a particular bacterial pathogen, one successfully abstracts the temporal nature of this finding, which would be pertinent in the diagnosis of rheumatic fever or post-streptococcal glomerulonephritis. Note that no explicit definition of "recent" is part of this representation; if for some reason one needed to distinguish infection 2 weeks ago from infection 3 months ago, this abstraction would not suffice. Thus, there is a disadvantage to this simplification. Nonetheless, CDSS which use implicit temporal abstractions seem to perform well for time-sensitive clinical cases.

2.2.2 Inference Engine

There are many ways of programming an inference engine. The inference engine is the portion of the CDSS that combines the input and other data according to some logical scheme for output. Users of the system do not usually know—or need to know—how the engine works to achieve the results.

One such scheme for an inference engine is the Bayesian network. Recall that Bayes' rule helps us express conditional probabilities—the likelihood of one event given that another has occurred. A Bayesian network is a way to put Bayes' rule to work by laying out graphically which events influence the likelihood of occurrence of other events. Figure 2.2 shows a Bayesian network for the diagnosis of pneumonia.

The arrows in the diagram indicate all of the conditional relationships between findings and diagnoses. Note that the symptoms listed are not necessarily independent; since febrile patients are often tachypneic, even in the absence of lung disease, one cannot say the two are as independent as Bayesian reasoning requires. Conceptually, this network simply states that the diagnosis of pneumonia is supported by the presence of three symptoms. The strength of association—that is, how strongly pneumonia is suggested by each of the three symptoms—varies with each symptom–disease pairing. By "activating" all three nodes (cough, fever, and tachypnea) the probability of pneumonia is maximized. Of course, each of these three nodes might be tied to other disease states in the knowledge base (like lung cancer or upper respiratory infection).

Bayesian networks can be complex, but their usefulness comes from their ability to represent knowledge in an intuitively appealing way. Inference engines that operate on the basis of a network simply adjust probabilities based on simple mathematical relationships between nodes in the network. Iliad [20, 29], an early CDSS, was one such program that was built on Bayesian reasoning, and whose reasoning engine can be described as a Bayesian network. Bayesian network systems have been designed and validated for a variety of clinical situations, including tumor classification, cancer prognosis, and ectopic pregnancy detection [30–33]. Inputs to these systems include data ordinarily found in the EHR, although none of these have found their way into commercial use.

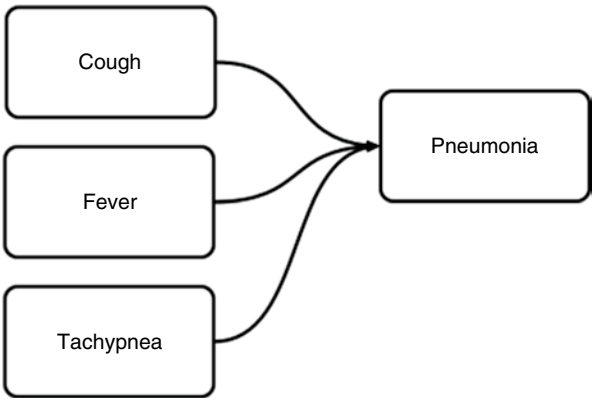

Fig. 2.2 A Bayesian network for the diagnosis of pneumonia

Production rule systems are another method of programming an inference engine. The rules of predicate logic dictate the functioning of such an engine as it combines statements to form new conclusions. MYCIN, described earlier, uses a production rule system. Production rules are an intuitively attractive way to start thinking about CDSS, since so much of the care physicians give in daily practice follows certain well known rules (e.g., giving patients with asthma an influenza vaccine each year). Other CDSS using production rules include Care Assistant, a general purpose, rule-based tool developed at the Childrens Hospital of Philadelphia that accepts input from an EHR via Web services and delivers decision support to the EHR user for immunizations and other treatment guidelines [34–37]. Chapter 5 describes this type of CDS via Web services in more detail. While this system was a customized add-on to the EHR, today some of this functionality is available in commercial EHR products, especially on the domain of immunizations.

An appealing solution to the problem of constructing inference engines in a clinical setting is to develop a cognitive model of actual clinical reasoning. In other words, one could study the reasoning that a physician uses and attempt to create a computerized version of that cognitive task. Workers in the field of artificial intelligence, in modeling human cognition, have developed the notion of "frames" or schemes, as a reasonable cognitive model. A frame consists of a set of "slots" into which fit details of a particular kind of information. For example, a disease frame may have a slot for etiologic agent and time course. Frames can be used to construct a semantic network model of the world, which may then be searched for answers to questions based on a particular situation. One such application of frames in medicine is the criterion-table method of diagnosing diseases like rheumatoid arthritis or Kawasaki disease. By applying a list of criteria, physicians can classify patients by diagnosis. The AI/Rheum system [38, 39] employed this familiar device in an inference engine that could have been used outside its original domain of rheumatologic diseases.

2.2.3 Knowledge Base

For CDSS to work, they must possess some form of medical knowledge. Obviously, the method of encoding this knowledge must match the inference engine design. For example, a CDSS based on a Bayesian network must contain probabilities—prior, conditional, and posterior—of diseases and findings. A big obstacle to building such a knowledge base is that many relevant probabilities are not known. While the medical literature can surely help with this task, and CDSS developers use the literature to varying degrees in building their knowledge bases, knowledge base developers must resort to estimates of probabilities, based on the clinical judgment of experts, to fill in the needed numbers. Unfortunately, physicians can exhibit markedly variable behavior in supplying such numbers, and probabilities can vary from situation to situation, even with the same disease entities (e.g., variations in disease prevalence with different populations).

Once one creates a knowledge base and populates it with some amount of data, the next task is to create a way to maintain it. Since many CDSS begin as funded academic research projects, it is no wonder that development of their knowledge bases often halts after the grant funds cease. Since knowledge base maintenance takes a tremendous amount of time, and since the market for some CDSS is rather small, many CDSS become too expensive to maintain. The knowledge-acquisition bottleneck [40] has been recognized as a problem in CDSS research.

2.2.4 Output

The output of CDSS is usually in the form of a list of possibilities, ranked in some order of probability. Sometimes probability is not the only criterion on which results are evaluated; for example, in the DXplain output, diseases which are not necessarily very likely, but whose misdiagnosis would be catastrophic, are flagged with a special disease-importance tag to call attention to the possibility [24]. Very often, physicians are not interested in the most likely diagnosis from a CDSS; for experienced physicians, the most likely diagnosis is obvious. It is the less likely diagnosis that one might fail to consider that interests physicians in CDSS, yet clearly it is difficult to draw the line between the rare and the ultra-rare.

2.3 Nonknowledge-Based Systems

The systems discussed so far have been knowledge-based in the sense that an expert must expressly encode medical knowledge into numerical form for the systems to work. The knowledge-based systems cannot simply “learn” how to do the reasoning

task from modeling human experts; the human expert must put the knowledge into the system explicitly and directly.

2.3.1 Neural Networks

There are systems that can learn from examples. Neural networks are the most widely recognized of these types of systems, and there are regular reports in the medical literature on their use in diverse fields [41–47].

Artificial neural networks are constructed in a fashion similar to biological neural networks. Neuron bodies ("nodes") are connected to one another by axons and dendrites ("links"). Nodes may be turned on or off, just as a biological neuron can be in an activated or inactivated state. Activation of a node causes activation of a signal on a link. The effect of that signal depends on the weight assigned to that link. In most learning neural networks, some nodes are input nodes and some are output nodes. In the CDSS context, the input nodes would be findings and the output nodes would be possible diseases. To understand how a neural network might work, consider the problem of determining whether a person with a sore throat has streptococcal infection (as opposed to a harmless viral infection). There are many input nodes to this decision, and perhaps two output nodes, strep infection and viral infection. By presenting to a neural network many thousands of cases of sore throat (where the outcome is known), the neural network would "learn," for example, that the presence of cough decreases the likelihood of strep, and the height of fever increases this likelihood.

The appealing feature of neural networks—and what separates this technique from other methods of discovering relationships among data, like logistic regression—is the ability of the system to learn over time. A neural network changes its behavior based on previous patterns. In a domain where the relationship between findings and diseases might change, like infectious disease surveillance, this changing behavior can be desirable. Another desirable feature of neural networks is the lack of necessity to understand complex relationships between input variables; the network learns these relationships as it changes the links between its nodes. This is the principal difference between neural networks and Bayesian networks. In the latter, one explicitly constructs the network based on one's knowledge of pathophysiology and known probabilities. With neural networks, the links are established as the network is developed, often on the basis of a learning process, without regard to pathophysiologic facts. A disadvantage of neural networks, however, is that unlike the other systems discussed, the "rules" that the network uses do not follow a particular logic and are not explicitly understandable.

2.3.2 Genetic Algorithms

Genetic algorithms represent another nonknowledge-based method for constructing CDSS. Genetic algorithms take their name from an analogy to the molecular rearrangements that take place in chromosomes. Genes rearrange themselves randomly; such rearrangements give rise to variations in an individual, which can affect the individual's ability to pass on genetic material. Over time, the species as a whole incorporates the most adaptive features of the "fittest" individuals. Genetic algorithms take a similar approach. To use a genetic algorithm, the problem to be solved must have many components (e.g., a complex cancer treatment protocol with multiple drugs, radiation therapy, and so on). By selecting components randomly, a population of possible solutions is created. The fittest of these solutions (the one with the best outcome) is selected, and this subpopulation undergoes rearrangement, producing another generation of solutions. By iteratively extracting the best solutions, an optimal solution can be reached. The main challenge in using genetic algorithms is in creating the criteria by which fitness is defined. Since the computing power required to use both genetic algorithms and neural networks is considerable, these techniques have had only limited use in medicine.

2.4 Model for Evaluating the Appropriateness of CDSS

In a technology environment dominated by electronic health record (EHR) systems [48–51] most of the decision support that clinicians today face comes in the form of alerts presented during the normal use of the EHR [52]. The mathematics behind these alerts is usually a straightforward application of conditional logic, e.g.:

If current order's medication is in Nephrotoxic-Drug-Group,
And Creatinine-Clearance > Age-Specific-Threshold,
Then Display warning "Use caution when prescribing nephrotoxic drugs in this patient."

Studies of this kind of decision support suggest that clinicians ignore such alerts at high rates [53, 54]. The usual explanation is "alert fatigue" [55] but more complex sociotechnical factors affect the impact alerts have on quality of care [56, 57]. In any case, simple alerting as a form of decision support in EHRs has been shown to be of limited effectiveness, in contrast to other sources of information that clinicians use to make decisions. The ways laboratory tests are evaluated may help us formulate a way to develop metrics for quantifying the physician response to alerts. For example, in the case of laboratory test results, the classic method of evaluation of this kind of clinical data is the 2×2 table, shown in Fig. 2.3.

In this case, one can calculate several metrics that can inform the clinical user about the performance of the test:

	Strep throat present	Strep throat absent
Rapid strep test positive	*a* True positive	*b* False positive
Rapid strep test negative	*c* False negative	*d* True negative

Fig. 2.3 Typical 2×2 table for a common laboratory test, indicating the four possible outcomes of applying the test in a population of patients, some of whom have the disease in question. The letters serve as a convenient way to refer to each cell of the table (see text)

	Condition present	Condition absent
Alert fires	*a* True positive	*b* False positive
Alert does not fire	*c* False negative	*d* True negative

Fig. 2.4 Possible 2×2 table for an alert embedded in an electronic health record. The alert would fire under the circumstances for which it was programmed, but typically the programming would not detect the condition perfectly. Again, the letters serve as a convenient way to refer to cells of this table

- Sensitivity (true positive rate among those with disease) = $a / (a + c)$
- Specificity (true negative rate among those without disease) = $d / (b + d)$
- Positive predictive value (PPV; true positive rate among positive tests) = $a / (a + b)$
- Negative predictive value (NPV; true negative rate among negative tests) = $d / (c + d)$

A highly sensitive test picks up a high proportion of those who have the disease. A highly specific test means there will be very few false positives. Using these metrics, one can judge the usefulness of a laboratory test as a decision support aid. For example, one could design guideline recommendations based on whether a test was highly sensitive (useful for ruling *out* disease) or highly specific (more useful for ruling *in* disease). There are other methods of using these metrics in the interpretation of laboratory test results that are beyond the scope of this chapter.

For EHR-based alerts, one might like to have a similar 2×2 table in order to make judgments about the usefulness of the alert (Fig. 2.4).

If one could gather data from one's EHR to fill in this table, one could gain an appreciation for the performance of an alert, and be able to make decisions about whether an alert was valuable. For example, one could deploy a highly sensitive alert to screen for a condition (in a manner similar to a lab test), but a more specific alert in a part of the workflow where condition confirmation is more important. In

theory, it is possible to gather data to fill in this table, but given that the manual chart review and data collection is expensive, one usually uses data from a report in which the EHR system displays when the alert fired (boxes *a* and *b* in 2.4). The EHR cannot report on box *c* (condition present but alert did not fire) since, had the EHR been able to detect the condition, those data would have gone to boxes *a* or *b*. In other words, it is a logical impossibility for a computer system to present a report on something it is not programmed to know. As a result, we are left with the ability to calculate only the true positive rate, and cannot obtain the other metrics without an infeasible amount of manual data collection. Reliance on true positive rate only—what one might call the "tyranny of box *a*" generates false contentment that a given alert is effective. The question of "how good is this alert?" often remains unanswered among the numerous safety-driven requests to add more alerts that must be accommodated in a typical EHR implementation.

2.5 Summary

Understanding clinical decision support systems requires a basic understanding of probability and logic. Set theory, familiar to most practitioners who have manipulated collections of literature citations in PubMed, provides the basis for understanding probability and other computational methods for reasoning. Probability—in particular, conditional probability—is the principle behind most modern CDSS, but non-probabilistic heuristic techniques have been used to good effect in the past.

Understanding CDSS can be facilitated by considering four basic components of the CDSS process: input, reasoning engine, knowledge base, and output. Input is often constrained by controlled vocabularies or limitations in temporal expression of clinical features. Reasoning engines take on different designs, but their operation is usually transparent to the user of a CDSS. Knowledge bases contain data from which the reasoning engine takes rules, probabilities, and other constructs required to convert the input into output. Output can take many forms, including a differential diagnosis list or simply a probability of a particular diagnosis. Nonknowledge-based systems use techniques of machine learning to generate methods of turning input into meaningful output, regardless of an explicit representation of expert knowledge. While very important to do, it is a challenge to develop appropriate metrics to judge the appropriateness of CDSS performance.

References

1. NCBI. PubMed: National Center for Biotechnology Information. 2015 [cited 2015 7/4/2015]. Available from: http://www.ncbi.nlm.nih.gov/pubmed/.
2. Schiff GD, Bates DW. Can electronic clinical documentation help prevent diagnostic errors? N Engl J Med. 2010;362(12):1066–9. doi:10.1056/NEJMp0911734.

3. Richardson JE, Ash JS. A clinical decision support needs assessment of community-based physicians. J Am Med Inform Assoc. 2011;18 Suppl 1:i28–35. doi:10.1136/amiajnl-2011-000119. PubMed PMID: 21890874, PubMed Central PMCID: PMC3241161.
4. Kuperman GJ, Bobb A, Payne TH, Avery AJ, Gandhi TK, Burns G, et al. Medication-related clinical decision support in computerized provider order entry systems: a review. J Am Med Inform Assoc. 2007;14(1):29–40. doi:10.1197/jamia.M2170. PubMed PMID: 17068355, PubMed Central PMCID: PMC2215064.
5. Brokel J. Evidence-based clinical decision support improves the appropriate use of antibiotics and rapid strep testing. Evid Based Med. 2014;19(3):118. doi:10.1136/eb-2013-101625.
6. Henderson EJ, Rubin GP. The utility of an online diagnostic decision support system (Isabel) in general practice: a process evaluation. JRSM Short Rep. 2013;4(5):31. doi:10.1177/2042533313476691. PubMed PMID: 23772310; PubMed Central PMCID: PMCPMC3681231.
7. Bond WF, Schwartz LM, Weaver KR, Levick D, Giuliano M, Graber ML. Differential diagnosis generators: an evaluation of currently available computer programs. J Gen Intern Med. 2012;27(2):213–9. doi:10.1007/s11606-011-1804-8. PubMed PMID: 21789717; PubMed Central PMCID: PMCPMC3270234.
8. Hoffer EP, Feldman MJ, Kim RJ, Famiglietti KT, Barnett GO. DXplain: patterns of use of a mature expert system. AMIA Annu Symp Proc. 2005:321–5. PubMed PMID: 16779054; PubMed Central PMCID: PMC1560464.
9. Gibofsky A. Clinical manifestations and diagnosis of acute rheumatic fever. In: Post TW, editor. UpToDate. Waltham: Wolters Kluwer Health; 2013.
10. Reggia JA, Nau DS, Wang PY. Diagnostic expert systems based on a set covering model. Int J Man Mach Stud. 1983;19(5):437–60.
11. Hsieh YZ, Su MC, Wang PC. A PSO-based rule extractor for medical diagnosis. J Biomed Inform. 2014;49:53–60. doi:10.1016/j.jbi.2014.05.001.
12. Leite CR, Sizilio GR, Neto AD, Valentim RA, Guerreiro AM. A fuzzy model for processing and monitoring vital signs in ICU patients. Biomed Eng Online. 2011;10:68. doi:10.1186/1475-925X-10-68. PubMed PMID: 21810277, PubMed Central PMCID: PMC3162941.
13. Miranda GH, Felipe JC. Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization. Comput Biol Med. 2014. doi:10.1016/j.compbiomed.2014.10.006.
14. Yu VL, Buchanan BG, Shortliffe EH, Wraith SM, Davis R, Scott AC, et al. Evaluating the performance of a computer-based consultant. Comput Programs Biomed. 1979;9(1):95–102.
15. Doherty J, Noirot LA, Mayfield J, Ramiah S, Huang C, Dunagan WC, et al. Implementing GermWatcher, an enterprise infection control application. AMIA Annu Symp Proc. 2006:209–13. PubMed PMID: 17238333; PubMed Central PMCID: PMC1839697.
16. Kahn MG, Steib SA, Dunagan WC, Fraser VJ. Monitoring expert system performance using continuous user feedback. J Am Med Inform Assoc. 1996;3(3):216–23. PubMed PMID: 8723612, PubMed Central PMCID: PMC116303.
17. Martone WJ, Gaynes RP, Horan TC, Danzig L, Emori TG, Monnet D, et al. National Nosocomial Infections Surveillance (NNIS) semiannual report, May 1995. A report from the National Nosocomial Infections Surveillance (NNIS) System. Am J Infect Control. 1995;23(6):377–85.
18. Bayes T. An essay towards solving a problem in the doctrine of chances. 1763. MD Comput. 1991;8(3):157–71.
19. Warner HR, Toronto AF, Veasey LG, Stephenson R. A mathematical approach to medical diagnosis: application to congenital heart disease. 1961. MD Comput. 1992;9(1):43–50.
20. Warner Jr HR, Bouhaddou O. Innovation review: Iliad – a medical diagnostic support program. Top Health Inf Manag. 1994;14(4):51–8.
21. Shortliffe EH, Davis R, Axline SG, Buchanan BG, Green CC, Cohen SN. Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system. Comput Biomed Res. 1975;8(4):303–20.

22. Vincent WR, Martin CA, Winstead PS, Smith KM, Gatz J, Lewis DA. Effects of a pharmacist-to-dose computerized request on promptness of antimicrobial therapy. J Am Med Inform Assoc. 2009;16(1):47–53. doi:10.1197/jamia.M2559. Epub 2008/10/28doi: M2559 [pii].
23. Miller RA. A history of the INTERNIST-1 and Quick Medical Reference (QMR) computer-assisted diagnosis projects, with lessons learned. Yearb Med Inform. 2010:121–36.
24. Elkin PL, Liebow M, Bauer BA, Chaliki S, Wahner-Roedler D, Bundrick J, et al. The introduction of a diagnostic decision support system (DXplain) into the workflow of a teaching hospital service can decrease the cost of service for diagnostically challenging Diagnostic Related Groups (DRGs). Int J Med Inform. 2010;79(11):772–7. doi:10.1016/j.ijmedinf.2010.09.004. PubMed PMID: 20951080, PubMed Central PMCID: PMC2977948.
25. Norman GR, Eva KW. Diagnostic error and clinical reasoning. Med Educ. 2010;44(1):94–100. doi:10.1111/j.1365-2923.2009.03507.x.
26. Poses RM, Anthony M. Availability, wishful thinking, and physicians' diagnostic judgments for patients with suspected bacteremia. Med Decis Mak. 1991;11(3):159–68.
27. Groopman J. How doctors think. Boston: Houghton Mifflin Publishing Co.; 2008.
28. Aliferis CF, Cooper GF, Miller RA, Buchanan BG, Bankowitz R, Giuse N. A temporal analysis of QMR. J Am Med Inform Assoc. 1996;3(1):79–91.
29. Lau LM, Warner HR. Performance of a diagnostic system (Iliad) as a tool for quality assurance. Comput Biomed Res. 1992;25(4):314–23.
30. Chen R, Wang S, Poptani H, Melhem ER, Herskovits EH. A Bayesian diagnostic system to differentiate glioblastomas from solitary brain metastases. Neuroradiol J. 2013;26(2):175–83.
31. Gevaert O, De Smet F, Kirk E, Van Calster B, Bourne T, Van Huffel S, et al. Predicting the outcome of pregnancies of unknown location: Bayesian networks with expert prior information compared to logistic regression. Hum Reprod. 2006;21(7):1824–31. doi:10.1093/humrep/del083.
32. Gevaert O, De Smet F, Timmerman D, Moreau Y, De Moor B. Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. Bioinformatics. 2006;22(14):e184–90. doi:10.1093/bioinformatics/btl230.
33. Sesen MB, Peake MD, Banares-Alcantara R, Tse D, Kadir T, Stanley R, et al. Lung cancer assistant: a hybrid clinical decision support application for lung cancer care. J R Soc Interface. 2014;11(98):20140534. doi:10.1098/rsif.2014.0534. PubMed PMID: 24990290, PubMed Central PMCID: PMC4233704.
34. Bell LM, Grundmeier R, Localio R, Zorc J, Fiks AG, Zhang X, et al. Electronic health record-based decision support to improve asthma care: a cluster-randomized trial. Pediatrics. 2010;125(4):e770–7. doi:10.1542/peds.2009-1385.
35. Fiks AG, Grundmeier RW, Biggs LM, Localio AR, Alessandrini EA. Impact of clinical alerts within an electronic health record on routine childhood immunization in an urban pediatric population. Pediatrics. 2007;120(4):707–14. doi:10.1542/peds.2007-0257.
36. Fiks AG, Hunter KF, Localio AR, Grundmeier RW, Bryant-Stephens T, Luberti AA, et al. Impact of electronic health record-based alerts on influenza vaccination for children with asthma. Pediatrics. 2009;124(1):159–69. doi:10.1542/peds.2008-2823.
37. Forrest CB, Fiks AG, Bailey LC, Localio R, Grundmeier RW, Richards T, et al. Improving adherence to otitis media guidelines with clinical decision support and physician feedback. Pediatrics. 2013;131(4):e1071–81. doi:10.1542/peds.2012-1988.
38. Athreya BH, Cheh ML, Kingsland 3rd LC. Computer-assisted diagnosis of pediatric rheumatic diseases. Pediatrics. 1998;102(4), E48.
39. Bernelot Moens HJ. Validation of the AI/RHEUM knowledge base with data from consecutive rheumatological outpatients. Methods Inf Med. 1992;31(3):175–81.
40. Musen MA, van der Lei J. Knowledge engineering for clinical consultation programs: modeling the application area. Methods Inf Med. 1989;28(1):28–35.
41. Azimi P, Mohammadi HR, Benzel EC, Shahzadi S, Azhari S, Montazeri A. Artificial neural networks in neurosurgery. J Neurol Neurosurg Psychiatry. 2015;86(3):251–6. doi:10.1136/jnnp-2014-307807.

42. Zhang F, Feng M, Pan SJ, Loy LY, Guo W, Zhang Z, et al. Artificial neural network based intracranial pressure mean forecast algorithm for medical decision support. Conf Proc IEEE Eng Med Biol Soc. 2011;2011:7111–4. doi:10.1109/IEMBS.2011.6091797.
43. Santhiyakumari N, Rajendran P, Madheswaran M. Medical decision-making system of ultrasound carotid artery intima-media thickness using neural networks. J Digit Imaging. 2011;24(6):1112–25. doi:10.1007/s10278-010-9356-8. PubMed PMID: 21181487, PubMed Central PMCID: PMC3222553.
44. Xie X, Wang L, Wang A. Artificial neural network modeling for deciding if extractions are necessary prior to orthodontic treatment. Angle Orthod. 2010;80(2):262–6. doi:10.2319/111608-588.1.
45. Larder B, Wang D, Revell A. Application of artificial neural networks for decision support in medicine. Methods Mol Biol. 2008;458:123–36.
46. Shimada T, Shiina T, Saito Y. Detection of characteristic waves of sleep EEG by neural network analysis. IEEE Trans Biomed Eng. 2000;47(3):369–79. doi:10.1109/10.827301.
47. Somoza E, Somoza JR. A neural-network approach to predicting admission decisions in a psychiatric emergency room. Med Decis Making. 1993;13(4):273–80.
48. Grinspan ZM, Banerjee S, Kaushal R, Kern LM. Physician specialty and variations in adoption of electronic health records. Appl Clin Inf. 2013;4(2):225–40. doi:10.4338/ACI-2013-02-RA-0015. PubMed PMID: 23874360; PubMed Central PMCID: PMCPMC3716415.
49. Hsiao CJ, Hing E, Socey TC, Cai B. Electronic medical record/electronic health record Systems of office-based physicians: United States, 2009 and preliminary 2010 state estimates. In: Statistics NCfH, editor. Washington, DC: Centers for Disease Control; 2010.
50. Jha AK, DesRoches CM, Kralovec PD, Joshi MS. A progress report on electronic health records in U.S. hospitals. Health Aff (Millwood). 2010;29(10):1951–7. doi:10.1377/hlthaff.2010.0502. hlthaff.2010.0502 [pii].
51. Leu MG, O'Connor KG, Marshall R, Price DT, Klein JD. Pediatricians' use of health information technology: a national survey. Pediatrics. 2012;130(6):e1441–6. doi:10.1542/peds.2012-0396.
52. Mann D, Knaus M, McCullagh L, Sofianou A, Rosen L, McGinn T, et al. Measures of user experience in a streptococcal pharyngitis and pneumonia clinical decision support tools. Appl Clin Inf. 2014;5(3):824–35. doi:10.4338/ACI-2014-04-RA-0043. PubMed PMID: 25298820, PubMed Central PMCID: PMC4187097.
53. Nanji KC, Slight SP, Seger DL, Cho I, Fiskio JM, Redden LM, et al. Overrides of medication-related clinical decision support alerts in outpatients. J Am Med Inform Assoc. 2014;21(3):487–91. doi:10.1136/amiajnl-2013-001813.
54. Slight SP, Nanji KC, Seger DL, Cho I, Volk LA, Bates DW. Overrides of clinical decision support alerts in primary care clinics. Stud Health Technol Inf. 2013;192:923.
55. Lo HG, Matheny ME, Seger DL, Bates DW, Gandhi TK. Impact of non-interruptive medication laboratory monitoring alerts in ambulatory care. J Am Med Inform Assoc. 2009;16(1):66–71. doi:10.1197/jamia.M2687. Epub 2008/10/28M2687 [pii].
56. Ash J, Sittig D, Campbell E, Guappone K, Dykstra R. Some unintended consequences of clinical decision support systems. AMIA Annu Symp Proc. 2007:26–30.
57. Ash J, Sittig D, Dykstra R, Campbell E, Guappone K. The unintended consequences of computerized provider order entry: findings from a mixed methods exploration. Int J Med Inform. 2009;78 Suppl 1:S69–76.