

Kabisa: an interactive computer-assisted training program for tropical diseases

J Van den Ende,[1] K Blot,[2] L Kestens,[3] A Van Gompel[1] & E Van den Enden[1]

[1]Institute of Tropical Medicine, Clinical Department, Antwerp, Belgium; [2]Medical Department, Pfizer, New York, USA; and [3]Immunology Department, ITM, Antwerp, Belgium

SUMMARY

In Europe, tropical pathology is usually taught in special short courses, intended for those planning to practise in developing countries. The theoretical knowledge to be assimilated during this short period is considerable, and turning such newly acquired knowledge into competence is difficult.

Kabisa is a computer-based training program for tropical diseases. Instead of concentrating on strictly tropical diseases, students are trained in recognizing diseases in patients presenting randomly in an imaginary reference hospital in a developing country. Databases are compiled by experts from experiences in various parts of Africa, Asia and tropical America. Seven languages and three levels of competence can be chosen by the student. Updating of all databases is possible by teachers who want to describe a particular setting. A 'consistency checker' verifies the internal consistency of a new configuration. The logical engine is based upon both a 'cluster' and a Bayesian logic, with built-in corrections for related disease characteristics. This correction allows calculated probabilities to stay closer to real probabilities, and avoids the 'probability overshoot' that is inherent to 'idiot Bayes' calculations. The program provides training in diagnostic skills in an imaginary second-line setting in a tropical country. It puts tropical and cosmopolitan diseases in perspective and combines applied clinical epidemiology and pattern recognition within varying sets of presenting symptoms. Students are guided in searching for the most relevant disease characteristics, in ranking disease probability, and in deciding when to stop investigating.

Keywords

Belgium; *computer-assisted instruction; computer systems; teaching, methods; tropical medicine, *education

INTRODUCTION

Theory and practice

Medical theory is almost invariably taught in disease-based, 'cause-leads-to-effect' relationships, i.e. students learn about a disease and its effect on the human body. This teaching approach is not easily turned into usable competence, since medical practice is essentially symptom-based and uses 'effect-points-to-cause' relationships. This discrepancy between theory and practice is a major problem for every new doctor. Before the advent of the microcomputer, attempts to bridge this gap between theory and practice were largely limited to time-consuming hospital-based case presentations. As the necessary hardware became available and affordable, computer-assisted learning (CAL) programs found their way into the teaching provision.

Teaching tropical medicine in Europe

In Europe, tropical pathology is usually taught in special short courses, organized for people who are planning to practice medicine in developing countries. These courses last between 2 weeks up to 6 months. The theoretical knowledge to be assimilated during this short period is completely new and quite large, and turning this newly acquired knowledge into competence is difficult. Since most students leave for the developing countries shortly after the course, a CAL program which simulates practical medical situations could provide useful instruction.

Objectives of Kabisa

Kabisa is a Windows 3.1 hosted tutor and training program that targets apprentices in tropical medicine as its user base. 'Kabisa' is the Kiswahili translation for 'hand in the fire, I'm sure that...'. The word refers to pattern recognition and threshold of certainty. The first DOS version emerged from a card game, also entitled Kabisa.

Its main objective is to improve the diagnostic skill of the users by helping them recognize disease patterns within varying sets of presenting symptoms, rank these diseases in likelihood, and decide when to stop investigating and to start treatment. A logical point-and-click interface and context sensitive 'help' function make Kabisa user-friendly, even to computer novices. The program consists of three modules: a simple text editor for

Correspondence: Dr J Van den Ende, Institute of Tropical Medicine, Nationalestraat 155, 2000 Antwerp, Belgium

taking notes while working; a training menu; and an optional database editor.

BACKGROUND AND LOGIC

Computer language and requirements

The program is developed using Borland C++ version 3.0, with Turbo C++ version 3.1 later on and depends heavily upon the object-oriented features of C++. Kabisa will run on any IBM compatible microcomputer which is able to run Microsoft Windows 3.1 (minimum configuration: 80386 CPU with 4 MB RAM and VGA display). A Windows compatible mouse and hard disk are required, and installing the program to the hard disk will use approximately 1 megabyte of disk space.

Environment and assumptions

It is assumed that trainees will work in a general hospital, somewhere between a district (local) hospital and a university hospital. Only patients who have been referred by another health service, or patients presenting at the emergency ward of the hospital are considered. Every year, 10 000 patients are seen in the outpatient department or in the ward. We suppose that every patient has one single disease, and imported diseases are not considered.

The medical decision training is presented as a medical visit, during which the user has to identify the randomly generated disease by checking the presence or absence of characteristics. Kabisa evaluates the suggested characteristics and gives continuous feedback to the user.

Databases

The system's data are made up of disease entities, disease characteristics and their associations, as expressed by their respective sensitivities and specificities. Since the ultimate aim of the program is to make a diagnosis, disease entities are commonly referred to as *diagnoses*. A *characteristic* is any type of information that might be helpful in diagnosing a disease. Symptoms, physical signs, laboratory results, X-ray and ultrasound are characteristics, but so are age and disease chronology.

Diagnoses

The diagnosis database contains about 250 diseases. The database is not restricted to tropical diseases: cosmopolitan diseases, such as pneumonia, measles, or myocardial infarction, have been added in order to approach more closely the reality of a consultation or a ward round in a tropical hospital.

Every diagnosis has a baseline (a priori) probability of occurring in the hospital setting. If a diagnosis is made in 200 of the 10 000 patients that are seen on a yearly basis, its prevalence rate is 0·02. Of course, this figure is many times greater than its corresponding prevalence rate in the general population. Since 'hospital prevalence' figures are not available in the literature, we performed a Delphi survey (an estimation by specialists) among a panel of three internists with several years of experience in a setting close to the one the program assumes. If this method yielded too variant prevalence rates, the results were discussed with other experts until a consensus was reached. All very uncommon diseases were given the same pre-test probability or prevalence rate (0·0001), in order to facilitate calculations and not to be too speculative.

Every disease is scored by its severity, treatability, cost of treatment, toxicity of treatment, and degree of stigma. These parameters are used to calculate a *diagnostic threshold* (test-treatment threshold), i.e. a probability level that must be reached before a diagnosis can be made (Pauker & Kassirer 1980). This corresponds to a common clinical practice: when suspecting a fatal, uncurable and stigmatizing disease such as AIDS, a clinician wants a high degree of confidence before concluding this diagnosis. If the disease is potentially fatal but curable when treatment is instituted promptly, e.g. acute appendicitis, the diagnostic threshold will lower considerably.

Characteristics

The 'characteristic' database contains approximately 240 characteristics. Each characteristic is defined by its prevalence and cost. A minimum estimate of a characteristic's prevalence can be obtained by multiplying each disease's prevalence rate by the sensitivity of the characteristic, and summing the obtained quantities, which are nothing else than the probabilities of a characteristic being generated by a certain disease. On top of this minimum estimate, a variable amount of 'unspecified' characteristic prevalences may have to be added: this is the proportion of a characteristic which cannot be explained by identifiable disease. If 10% of the patients cough by explainable causes, a lot more will cough without identifiable disease. For abdominal guarding in most cases a cause is found, hence the characteristic prevalence rate is not much higher than the explainable prevalence rate. All obtained characteristic prevalences were functionally audited by a panel of internists for consistency within the program. If the characteristic is a symptom, it may be flagged as a 'presenting symptom' if it concerns a com-

plaint the patient may spontaneously evoke when consulting for his/her problem.

Associations

The association database links symptoms and diagnoses (diseases) together. Each characteristic–diagnosis relation is characterized by the sensitivity of this characteristic with regard to the given diagnosis.

Usually, probabilistic characteristic–diagnosis links are expressed in terms of both sensitivity and specificity. Sensitivities can often be found in the literature, otherwise they can be reasonably estimated by a panel of experts. Sensitivities can be assumed to be relatively identical over different geographic settings, as long as accessibility to health services is similar. The proportion of patients presenting with pain irradiating to the left arm in case of myocardial infarction is not likely to vary among different continents, but a substantially higher number of women in developing countries may present with an open ulcer when first consulting for breast cancer.

Specificities, on the contrary, are quite problematic to obtain. More importantly, a characteristic's specificity depends upon the epidemiological situation. A plasmodium-positive thick smear in a traveller returning from a tropical country will never be discarded as not meaningful, but the same finding could be quite normal in a resident of a malaria-endemic area. Even if this variation could be taken into account, directly estimating specificities remains a very technical and unwieldy task. However, specificities for a characteristic C and a disease D can be indirectly estimated if we know the prevalence rate of D, the sensitivity of C with regard to D, and the prevalence of C.

For age, sex and temporal delay, rather speculative likelihood ratios have been attributed to associations, since strict weights for these arguments are often not available.

Intercharacteristic relations are the common cause of overestimation of post-test probability, the so-called idiot Bayes. Estimating the real proportion of characteristic interdependence is extremely difficult. Two kinds of links allow to avoid this bias: on one side the 'include' relationship, which means that a characteristic is always present when an index characteristic is already listed; on the other hand, the 'possible' link. No efforts were made to specify these relations per disease. Notwithstanding these corrections, we still remain with an overestimation, therefore all likelihoods can be turned down by a constant factor in the Kabisa.ini file. We opted for this manual adaptation because students can be allowed a certain overestimation in the beginning of the training, according to the teacher's appreciation of progression.

The teacher's version allows modification of the databases in all respects so Kabisa can be adapted to another restricted geographic setting (the consistency checker warrants internal consistency of this new set of data).

Logical engine

Bayesian logic. The logical core of the program is an enhanced model of the so-called 'idiot Bayes' algorithm. In a Bayesian model, post-test odds for a disease are obtained by multiplying the pre-test odds of the disease with the positive likelihood ratio (LHR+) of a present characteristic, or with the negative likelihood ratio (LRH–) of an absent characteristic. This way, LHRs of several characteristics can be multiplied to give the final probability of a given disease. Kabisa uses this technique to keep the current probabilities for all 240 diseases of the database up-to-date during the 'medical visit'. The major flaw of this algorithm is that it will only work correctly when all characteristics in the model are independent. If interdependent characteristics (e.g. fever, headache, muscle pain, and shivering) are present, an 'idiot' multiplication will result in enormous probability overshoots. Hence, corrective mechanisms were designed in order to keep the calculated probabilities within reasonable limits.

Corrections to the Bayesian model. The first algorithm is based upon the recognition of characteristics' interdependence. Kabisa allows characteristics to be marked as 'linked' (cfr databases). This means that the presence of one characteristic may influence the likelihood ratio of another characteristic. For example, fever, muscle and joint pain are often seen together in the setting of a generalized infection. In the African database, fever and muscle pain have positive likelihood ratios (LHR+) of 1·3 and 3·2 for malaria, respectively. In idiot Bayes, the presence of both characteristics would lead to a multiplication of the malaria odds by a factor of 4·16. Joint pain (malaria LHR+: 3·2) is often seen together with muscle pain in this setting. If joint pain was present also, idiot Bayes would multiply the malaria odds by 13·3. To cope with this overshoot, Kabisa only uses the highest LHR+ when linked characteristics are found together. In our example, this would reduce the total odds multiplication from 13·3 to 3·2.

A second corrective algorithm uses the concept of 'alarm' characteristics. These are characteristics that alert the doctor to the possibility that something very serious is going on. Classical examples include neck stiffness, abdominal guarding, jaundice, acute mental disturbances, etc. If such a characteristic is unequivocally present, any doctor will stop considering diagnoses known not to present such characteristics. This means that odds

for diseases not related to the alarm characteristic are set to zero odds when an 'alarm' characteristic appears.

A third mechanism consists of correcting the weight of each LHR by reducing it with a constant factor, e.g. 0·8. This weighing factor can be defined in the Kabisa.ini file, which is a standard, editable Windows initialization file (when the program is started, the computer loads all constants defined in the ini file in the equations and algorithms).

Pattern recognition. In addition to this Bayesian logic, Kabisa uses a model of pattern recognition. Usually, at the beginning of a medical visit, odds are low for nearly all diseases in the database, especially when the presenting characteristics are commonly encountered (cough, fever, etc.). Hence, probabilistic reasoning will not be efficient in differentiating diagnoses at this point. Instead, doctors use pattern recognition to quickly 'browse' through the list of diagnoses which are known to be associated with the first characteristics that appear. For instance, if a patient presents with a cough and high fever, a wide range of diagnoses will be compatible with this pattern, including many types of respiratory tract infections as well as uncommon diseases like leptospirosis and Loeffler's disease. At this point, a doctor is able to exclude many of these diagnoses by looking for information in a directed way. If haemoptysis and substantial weight loss are present, the doctor will certainly be alerted to the possibility of pulmonary tuberculosis, which at this point will be the only diagnosis matching all four characteristics. Generally, very few questions or examinations will be needed to obtain a substantial differentiation in the odds of the diagnoses.

Therefore, pattern recognition is initially used to test whether the user is 'thinking in the right direction'. However, once one or more diagnoses reach a predefined probability, Kabisa will include Bayesian logic to evaluate the user's input. This way, the user may consider characteristics related to the diagnoses that do not perfectly match the pattern of presenting symptoms.

Characteristic input evaluation. When the user checks the presence of a new characteristic, the tutor will compare it both with disease patterns and disease probabilistic data. According to this evaluation, the user learns whether or not his/her input is considered adequate. If the user proposes a characteristic that is not helpful in finding the diseases that match the already given characteristics or the diseases that rank high in probability with the given characteristics, the tutor may ask the user which diagnosis he/she is considering. This diagnosis is then evaluated according to its current probability, showing the user which characteristics are and are not considered compatible with the proposed diagnosis. The minimal disease probability to be reached for accepting a characteristic as 'relevant' can be set in the ini file. We opted for an extensive reporting of negative general examinations for didactic purposes. If the student performs a chest auscultation, the tutor will consecutively report that there are no cardiac murmurs, no crepitations, no rales, no rub, no silence. The student knows, therefore, what he/she can expect from that examination.

Thresholds. If students are allowed to make a diagnosis on too weak grounds, it is possible that they may miss another diagnosis that is more likely. Clinicians always have to corroborate their hypothesis to a certain level of certainty. We assume that, for rural Africa, a treatment for meningitis can be started at a lower level of certainty than the level required to give the diagnosis of AIDS.

When the trainee points to a *diagnosis*, the tutor will accept it as a good differential diagnosis if it reached a certain level of post-test probability, corresponding to a specific threshold or diagnostic cut-off for this disease. This means that a diagnosis is sufficiently probable to start therapy, to refer the patient, or to communicate the diagnosis to the patient. We take into account disease severity, treatability, treatment toxicity and cost, risk for the community, and stigma for the patient. The tutor will agree with the actual diagnosis only if it reaches the threshold and if it is the most likely diagnosis (no other diagnoses should rank higher in probability). The baseline cut-off for considering and making a diagnosis (diagnostic threshold) can also be set in the ini file.

FUNCTIONAL DESCRIPTION

The tutor module

The tutor module is the main part of the program (Fig. 1). It generates a random diagnosis and presents two symptoms to the user, together with age characteristics (baby, child, adult, elderly person), sex and chronology (acute, subacute, chronic). It is up to the user to find out which disease was generated.

All relevant information (characteristics and diagnoses) is shown in listboxes. The user gathers evidence by checking characteristics. By double-clicking a characteristic listbox, the user 'asks' whether a characteristic is present or not (alphabetical retrieval is provided). Some symptoms and signs are general and should be specified: the tutor will automatically give a more detailed characteristic if it is present in the given disease. Accidentally present characteristics, not typical for a given disease, are not reported in order not to confuse the student. All characteristics the tutor will give or accept can be present with a given disease (perhaps they are not very interesting, but textbooks describe them). The tutor evaluates the user's input on the spot: when a characteristic is compatible

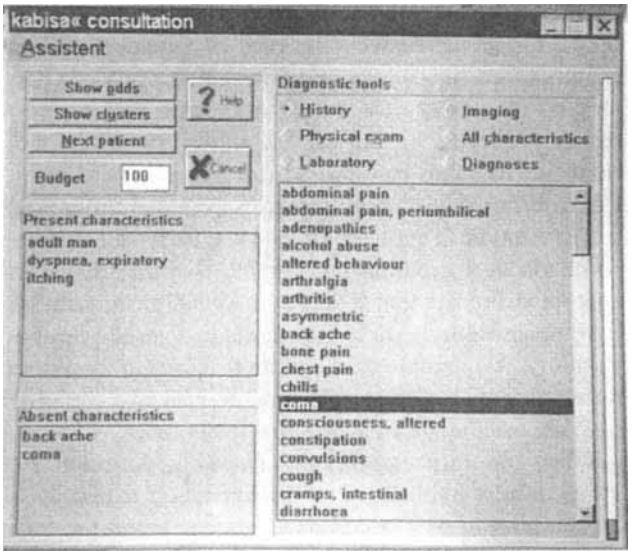

Figure 1 The consultation screen, showing the present and absent characteristics.

with a diagnosis, it is accepted and the user is congratulated for the finding. If the characteristic is incompatible with the generated diagnosis, the tutor will evaluate the input both through pattern matching and Bayesian logic: if the input is intelligent, the tutor will list the characteristic as absent. If the input is not to the point, the tutor will ask what diagnosis the user is considering. If the user wishes to confront his/her opinion with the system's data, Kabisa will explain which characterstics are compatible with the user's differential diagnosis and which are not.

Medical work in developing countries has always been associated with budget constraints. We chose to integrate this reality into Kabisa by providing the user with a budget at the start of his/her medical visits. Taking a medical history or doing a physical examination is considered to be free, but laboratory tests and imaging techniques have a price. When the user asks for a 'costly' examination, the tutor will tell him/her how much it will cost. The student can then decide not to perform the examination. Once the budget is exhausted, it becomes impossible to ask for more laboratory tests or imaging.

At some point in time, the user has to come to a diagnosis, which is only considered valid if sufficient evidence is present to support it. The diagnostic threshold (probability cut-off point) must be reached. The user's diagnosis is checked against Kabisa's internal list of differential diagnoses. Four possibilities are considered:

1. The user suggests a diagnosis that does not reach its threshold (yet). The tutor will explain why the diagnosis is considered inappropriate.
2. The user finds an acceptable differential diagnosis which is above its diagnostic threshold but which is not 'the' diagnosis. The finding of such a diagnosis is rewarded, and the user is encouraged to go on.
3. The user finds the computer-generated diagnosis, which is above its diagnostic threshold, but another diagnosis is (or other diagnoses are) more probable. The user is encouraged to obtain more evidence by adding characteristics, or to diminish the probability of the other diagnoses by looking for absent characteristics for these diagnoses.
4. The user finds the computer-generated diagnosis, it is above its diagnostic threshold and no other diagnosis is more probable: the user is congratulated, and the session will terminate.

Once the user finds the diagnosis, all the associated characteristics are displayed, along with their LHRs. At this point, the user may choose to see the 'probability history', which will display the consecutive probabilities after adding each characteristic for any disease the user wishes to evaluate. This may allow the users to critically review their hypotheses after the session. Every probability history may be logged to the editor screen, allowing printing or saving to disk.

During the medical visit, the trainee has two ways of viewing which differential diagnoses are currently considered by Kabisa. The 'differential diagnoses by clusters' only takes into account clusters of present characteristics. This means that it will only show diagnoses that are compatible with all of the present characteristics. It does not take into account that some characteristics may be listed as absent. For example, if malaria is part of the cluster of differential diagnoses, the absence of fever will not remove it from this list, but it *will* substantially reduce the probability of malaria in the next list!

The 'differential diagnoses by probability' shows the 'Bayesian' probability ranking of all possible diagnoses. An exclamation mark indicates that a diagnosis is above its threshold. This system takes present as well as absent characteristics into consideration when calculating the probability.

Additional training modules

Lecture. The dialogue box allows browsing through the associations that are defined between characteristics and diagnoses. A single click on one of the listed items allows the trainee to play to and fro between diagnoses and their associated characteristics. The number of diagnoses listed is dependent upon the trainee's user level settings, as defined in the options. With the log button,

the trainee can display his/her choices on the background screen and save or print them afterwards.

Extended lecture. This dialogue box gives the trainee an extended view of all relations between characteristics and diagnoses: prevalence rates, sensitivity, specificity, and positive and negative likelihood ratios. A two-by-two table shows the importance of each associated characteristic and diagnosis in the epidemiological context of the trainee's consultation. It displays the positive and negative predictive value for one characteristic, for the setting the trainee has chosen, and starting from the basic prevalence of the disease. The *'characteristic control'* shows how the prevalence of a given characteristic is distributed over all diagnoses, and allows the viewing of which proportion of jaundice is due to cholecystitis.

Shared characteristics. This module allows a comparison between different diseases. Characteristics in common and different characteristics are listed. Since the dictionary of characteristics is limited, some discussion might arise. This module gives merely orientation, not verdict. It may, however, be quite helpful during training sessions to identify characteristics that differentiate between two diagnoses that share many characteristics.

The former three modules are accessible during the medical visit session, but not during examinations (see below).

Clusters. This dialogue box lets the trainee experiment with characteristic associations. It immediately shows all diseases that may possibly match the characteristics the trainee enters. It is not possible to include absent characteristics.

Expert. With the aid of this feature the trainee can ask a probability ranking of diagnoses from present and absent findings in a given patient, starting from the prevalence in the imaginary hospital. It can be of help when the trainee is in an isolated setting. The trainee is, of course, limited to the available dictionary of characteristics.

Tools and options

When the databases are changed, the tutor will help to check if the system is still internally consistent. He/she will check if all new related data are completed. If, for instance, the prevalence of HIV infection is drastically changed, the specificity of a lot of characteristics will have changed. The consistency checker allows a quick look at errors that would have been created.

During the medical visit session, scores are stored for every action the user takes. This gives the trainee the score he/she accumulated over different consultation sessions. It also calculates the mean cost per session. All scores are set to zero when the program starts up and are kept during the whole Kabisa session. The score is based on the number of correct diagnoses and the number of irrelevant questions compared to a standard student group at the Institute (for details, see Appendix 1).

The exam module gives the trainee a series of consultations which cannot be interrupted. The number of patients to be seen, and the total budget for the session is set in the ini file. All tutor-help functions are blocked. At the end of the session, the tutor will give the trainee the final score and ask the trainee to insert a floppy disk in order to store the trainees' examination and score.

Since the program needs to be useful for a wide variety of users, several competence levels were designed.

- *junior doctor:* will show common diagnosis and will present simple consultations;
- *senior doctor:* will show all possible diagnoses and will present more difficult consultations. However, the trainee will always see more difficult diagnoses in the list than the tutor is allowed to start up;
- in the basic version, to a certain question, the tutor will confirm the presence of all reliable symptoms and signs: this is not the real situation and explains why the trainee can reach such high posterior odds. In the version *Sensitivity Ruled,* just like in everyday clinical work, presence of symptoms and signs will follow sensitivity: if this is 20%, the tutor should answer yes 20% of the time guided by its random generator.

Three geographic settings have already been worked out: Africa, Asia, and tropical America. By creating a new field in the diagnosis and characteristics database, the Kabisa logic can be applied to any other setting (e.g. gynaecology in Seattle).

For special purposes, some didactic tools were added:

- the trainee can start up a specified diagnosis, along with the symptoms he/she likes. This will allow him/her to work on differential diagnoses around this specific disease;
- all details of a consultation are stored in a file. The trainee can specify any file on their computer. This feature is interesting for examination sessions;
- the trainee can choose in which way probabilities are presented. The program will automatically recalculate odds to probability rates or to $\log_{10}$ odds.

The editor module

In the editor version, teachers can change the databases or build a database for a specific setting (Fig. 2). Diseases, prevalences, characteristics, associations between diseases and characteristics, sensitivities, and links between characteristics can all be updated and stored.

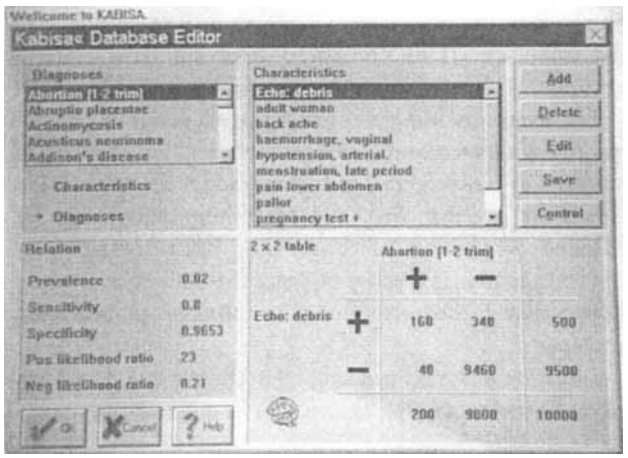

Figure 2 The editor screen. Teachers can change the databases or build a database for a specific setting. Diseases, prevalences, characteristics, associations between diseases and characteristics, and sensitivities can be updated and stored. Specificities are computer generated.

DISCUSSION

Didactics

Of course, it would be better to train students in real hospitals in a tropical setting, but scarcity of hospitals with an acceptable level of medicine; heavy workload of doctors in charge who are not available for time-consuming explanation; low training capacity of most doctors; geographic restriction in diagnoses; scarcity of rare diagnoses over a short training period; high travelling expenses all make a 'pre-training', or a 'dry-training' interesting.

The aim of the program is to put tropical and cosmopolitan diseases in perspective: students should be able to find whatever diagnosis starting from symptoms and signs, not only of tropical diseases. Experience has shown that students can have difficulties and mix up diseases they already know with a set of new diseases.

We also favour a combined training of tropical pathology and applied clinical epidemiology. Instead of giving boring lectures on Bayes' theorem, we start from application in everyday life. In the same way, we teach the weight of characteristics while explaining, e.g. typhoid fever. Kabisa has been written in the same spirit: disease presentation is continuously mixed with applied clinical epidemiology, and the discussion by the tutor encompasses both disciplines.

Kabisa is a training program, not an expert program. It has not been validated as an expert system. Bayesian-like logic is used to interpret the user's input and to provide sensible feedback. Nonetheless, the program could be of help in an isolated setting, in pointing to some diseases the trainee does not recall or has never seen before. An expert system differs from a training system in two respects: an expert system requires a complete library of disease characteristics which makes databases extremely large; and accuracy of sensitivity and specificity should be high, which is impossible for so many tropical settings. A training system requires continuous steering and discussion with a tutor: every step the trainee makes should be evaluated and discussed. This objective takes a major part of our software.

The program is sold at an affordable price, covering administration costs and hardware (US $50). The software is public domain and its development was supported by the Institute of Tropical Medicine, Antwerp. Medical schools who incorporate it in their teaching are asked for a certain amount for sharing costs of future developments.

Environment and assumptions

Copying or analysing a real geographical setting would restrict the number of diseases and could highly bias the training: a university hospital in Rwanda would give a quite different view compared with a small district hospital in Western Africa. Therefore we opted for a standard imaginary hospital per continent. We created a diagnosis mix for every continent. Every mix represents the average incidence rate of diagnoses in several hospitals in different locations throughout a continent. The student, therefore, will see East African and West African sleeping sickness together, which is a little unusual. Only referred patients are considered, avoiding numbers of people directly entering the reference level. This could bias the mix of diseases towards a first line. Moreover, logic is quite different on a secondary and tertiary level of medicine, where more energy and money can be spent in order to reach a final diagnosis, whereas medicine at the first level is more complaint-centred.

In simulating reality, we favour limiting presentation to real 'presenting symptoms'. Physical signs and other characteristics should be asked for by the trainee. In this respect, it is different from clinicopathological conferences, where an extensive case report is presented. Our exercise comes closer to the didactics of 'clinical problem solving' as presented in the *New England Journal of Medicine* (Kassirer 1995).

For didactic reasons, we include a few other assumptions. One patient has one disease, but classical complications of a given disease are considered; accidentally present characteristics, not typical for a given disease, are not reported (forgoing the clinical skill of filtering relevant data); in the basic options, accepting a characteristic as 'present' is not guided by sensitivity.

Databases

The database contains some rare diseases the tutor is not allowed to simulate. This is a logical consequence of the intention to evaluate the validity of a question. A trainee who thinks of a very rare disease should not be punished. Since the tutor checks all questions for relevance in the context of diseases, he/she would punish the trainee if he/she cannot find the disease the trainee is thinking of. Disease prevalences were obtained by the Delphi method: experience showed that there are few discrepancies among experts, if one thinks in orders of magnitude. Sensitivities do not vary so much between settings: we relied upon the literature or on expert opinion. Specificities, on the other hand, vary tremendously, depending on the mix of other diseases that cause the considered characteristic. Therefore we prefer computing specificities from disease prevalences, sensitivities and a factor depending on common presence of the characteristic.

Logic

The logical engine is both 'cluster' and 'Bayes' based. 'Cluster' logic takes into account only present characteristics, and does not provide a ranking based on probability. At the start of a consultation, clinicians rely on their 'pattern recognition', at the end they perform a probabilistic validation (Kassirer 1983). Both are available throughout the simulation.

In a perfect Bayesian model, the sum of the post-test probabilities of all diseases considered should never exceed 1 (or 100%). This premise is not fulfilled in our program: on one hand our data are generated by Delphi method and computer calculations, on the other hand the tutor always accepts characteristics regardless of sensitivity (in the basic options), which generates overshooting of post-test odds. In the more difficult option 'sensitivity ruled' we come closer to this premise. Rather than concentrating on Bayesian consistency, we focused on didactics, i.e. discussion of the relevance of each question or test ordered, fostering awareness of alarming characteristics and of budget constraints, and extensive reporting of negative examinations.

Making the diagnosis too early is one of the pitfalls in actual clinical work. Therefore, we included two barriers to diagnosis: first the trainee should reach a threshold; and second, no other disease should be more likely. The threshold differs somewhat from the classical test-treatment threshold described by Pauker & Kassirer (1980) as we consider more parameters and a diagnosis without treatment. Training concerning the lower or 'test threshold' is impossible in our program since the basic premise is that there is always a disease (disease-centred program).

Future

A project for a multimedia application of Kabisa is ongoing. Presenting symptoms can be linked with photographs; laboratory results can be shown as slides; some special features (tabetic gait, e.g.) can be shown in short moving sequences; and basic knowledge of a given disease can be presented *in extenso* with text, images and audio (Longstaffe *et al.* 1989).

A computer program that will focus on other aspects of diagnostic skills is currently under development in a complaint-centred program.

REFERENCES

Kassirer J P (1983) Teaching clinical medicine by iterative hypothesis testing. *New England Journal of Medicine* **309**, 921–3.

Kassirer J P (1995) Teaching problem-solving. How are we doing? *New England Journal of Medicine* **5**, 1507–9.

Longstaffe J A, Wittlestone M, Harkin P & Williams L (1989) Interactive video without tears: the anatomy of tutorial production for interactive video. *Medical Education* **23**, 307.

Pauker S G & Kassirer J P (1980) The threshold approach to clinical decision making. *New England Journal of Medicine* **302**, 1109–17.

Received 7 *February* 1996; *editorial comments to authors* 13 *March* 1996; *accepted for publication* 2 *April* 1996

APPENDIX 1

1) Formula for cut-off

If each of the parameters are scaled from 0 to 3 then the formula is:

cut-off = basic level, – severity/3 – treatability/3 + toxicity/6 + cost/6 + stigma/3.

2) Formula for score: $n = 20*D/S - (G/S - SG)/2 - (B/S - SB)$

S = consultations
D = diagnoses found
G = good questions
B = irrelevant questions
SG = Antwerp standard minimal good questions
SB = Antwerp standard minimal bad questions

Example: Suppose

S = 20, D = 10, G/S = 3, B/S = 1, SG = 3, SB = 1

then

10 – (3–3)/2 – (1–1) = 10/20