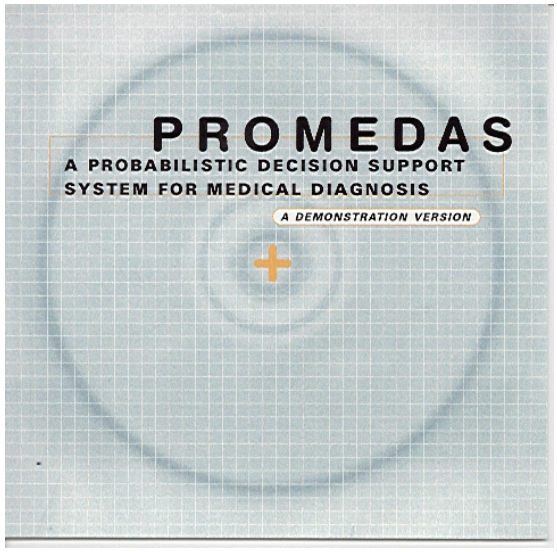
”PROMEDAS”
a probabilistic decision support system for medical diagnosis



SNN
Foundation for Neural Networks
Nijmegen, The Netherlands

UMCU
University Medical Centre Utrecht
Utrecht, The Netherlands

Summary

The use of patient-specific Decision Support Systems (DSS) may improve the quality and efficiency of health care, while reducing its costs at the same time. The adoption of such a system is largely compatible with the principles of ”Evidence Based Medicine” and patient oriented care.

PROMEDAS (PRObabilistic MEdical Diagnostic Advisory System) is a prototype DSS, based on a probabilistic model and advanced computational techniques. The system offers patient specific diagnostic advice. It presents a differential diagnosis and it supports the diagnostic process by indicating the most useful next step in the diagnostic process.

The system is intended to support diagnosis making in the setting of the outpatient clinic and for educational purposes. Its target-users are general internists, super specialists (i.e. endocrinologists, rheumatologists), interns and residents, medical students and others working in the hospital environment.

Currently, PROMEDAS is a stand alone application. In the future PROMEDAS may be integrated with a Hospital Information System and an Electronic Patient Record. This will facilitate its use in practice, and may augment its acceptance.

PROMEDAS is based on medical expert knowledge, acquired from the literature by the medical specialists in our project team. The acquired knowledge is stored in a database, in such a way that extension and maintenance of the expert knowledge is facilitated. Currently, the database covers large parts of endocrinology and lymphoma diagnostics. In near future, parts of vasculair medicine will be entered as well. From (parts of) this database, Bayesian network and an interface for PROMEDAS is automatically compiled. The network is the underlying model of PROMEDAS. Bayesian inference is used to query the system.

The PROMEDAS project is funded by STW (project nr: NNN5322). The goal of the project is to demonstrate that an accurate diagnostic DSS covering a large diagnostic repertoire in internal medicine is possible. The key technical innovation is the use of advanced approximate inference methods which allow Bayesian inference to be applied to large problem instances (patent submitted).

Contents

1 Introduction

A medical diagnostic Decision Support System (DSS) may be extremely useful because it is able to improve the accessibility of expert knowledge and patient information, resulting in quality improvement of the diagnostic process, increase of efficiency and reduction of costs. However, up to now, these systems have not yet entered daily clinical practice for a variety of reasons.

Why use decision support in medicine?

Medical diagnosis in modern medicine is a very complex process, requiring accurate patient data, a profound understanding of the medical literature and many years of clinical experience. Often, a clearcut diagnosis cannot be made and several alternatives must be considered, given the available patient data and the known medical mechanisms. As a result of this uncertainty, the decisions made by different physicians at different stages of the diagnostic process do not always agree and lack ”rationalisation” [1, 2].

The body of potentially useful knowledge that is relevant to even a relatively narrow diagnostic area may be too large to make the optimal (diagnostic) decision on the spot. In addition, modern information technology (especially through the Internet) increases the amount of available knowledge even more, probably further complicating this situation. Moreover, individual patients need ”individualised” decisions, because their characteristics differ from the ”average” and because of their individual wishes[3]. Individualisation of general research results to individual patient cases is cumbersome and time consuming,

The situation is particularly problematic for internal medicine because it covers an enormous range of diagnostic categories. As a result, internal medicine is differentiated in super specialisations and diagnosis often requires the decision by a team of super-specialists.

A medical diagnostic Decision Support System (DSS) is a computer program that contains all relevant medical domain knowledge about a certain medical domain and

generates a differential diagnosis on the basis of individual patient findings. In addition, a DSS can often suggest additional laboratory tests that are expected to be particularly informative to establish or rule out any of the diagnoses considered.

It is clear, that the benefits of a succesful DSS for internal medicine are far reaching. It will be appreciated by all medical specialists because it can give valuable information support for those patients that suffer from a disease outside his or her own super-specialization. It will result in an improved and more rationalized diagnostic process, as well as higher efficiency and cost-effectiveness.

What are the problems in current decision support systems?

The currently available systems (e.g. Meditel [5], QMR [6], Dxplain [7] and Iliad [8]) have not yet been very successful. Certainly their use is still not widespread and not established in daily routine. A variety of reasons may be responsible for this:

Lack of accuracy

Current systems that intend to cover a broad medical domain lack diagnostic accuracy due to the rather coarse level of detail of diagnostic categories (e.g. ICD-9 or ICD-10). As a result, the diagnoses made by these systems are of limited clinical relevance [9, 10]. Current systems that are used in practice are restricted to a relatively narrow field [11, 12]. Crucial problems with a detailed modeling of a large medical domain are 1) the maintenance of consistency and correctness of the medical knowledge model and 2) the intractability of computation. In the next sections we will discuss these problems in detail and the PROMEDAS approach to their solution.

Lack of EPD

Patient specific decision support needs input data from several sources. Integrated patient data in a common electronic patient dossier (EPD) is not available in most hospitals. For those subsystems that have been realized, the data can be incomplete and/or unreliable [14].

One of the problems is that different departments in internal medicine use their own terminology and definitions to characterize diagnoses. A common system for the whole of internal medicine requires a common terminology. Available diagnostic classification systems are either at a coarse level and therefore lack the required detail [15, 16], or specialised [17] and therefore too limited to meet the needs for a DDS covering a broad domain.

User acceptance

In the era of evidence based medicine the advice of “a black box” is unacceptable. An advice must be ’explained’, preferably on the basis of medical research literature. Currently available systems do not provide such eplanation.

PROMEDAS

In conclusion, modern medicine will benefit from computerised decision aids both to meet its own high standards and to keep pace with the stage of development in other domains such as manufacturing or the services industry. We strongly believe that a diagnostic DSS for a broad medical domain is viable and, eventually, marketable.

To avoid a ”gold rush style” in the search for these tools, the foremost thing to do is the development of safe and sound methods. The expertise of our multidisciplinary group primarily focuses on the following main parts of the methodology typically needed in the development of decision support tools:

- *Modelling and inference algorithms that are able to deal with large complex systems (section 2).*
- *Knowledge modelling in the medical domain (section 3).*
- *Evaluation and user aspects (section 4).*
- *Software development (section 5).*

2 Medical diagnosis and probabilistic modeling

A diagnostic decision support system offers diagnostic advice for a diagnostic problem regarding an individual patient. The system needs a representation of medical knowledge, i.e. a model, and it must be able to reason (i.e. compute) with patient specific data on the basis of this model. In the first part of this section, we discuss why probabilistic models are typically well suited for the representation of medical knowledge and for reasoning with this knowledge. A potential problem is the scalability of this approach to large knowledge domains. In the second part, we introduce principled approximation methods to deal with this issue.

Rule-based models versus probabilistic models

The process of medical diagnosis

Medical diagnosis is a process, by which a doctor searches for the cause (disease) that best explains the symptoms of a patient. The search process is sequential, in the sense that patient symptoms suggest some initial tests to be performed. Based on the outcome of these tests, a tentative hypothesis is formulated about the possible cause(s). Based on this hypothesis, subsequent tests are ordered to confirm or reject this hypothesis. The process may proceed in several iterations until the patient is finally diagnosed with sufficient certainty and the cause of the symptoms is established.

A significant part of the diagnostic process is standardized in the form of protocols. These are sets of rules that prescribe which tests to perform and in which order, based on the patient symptoms and previous test results. These rules form a decision tree, whose nodes are intermediate stages in the diagnostic process and whose branches point to additional testing, depending on the current test results. The protocols are defined in each country by a committee of medical experts.

Computerized medical diagnosis

The use of computer programs to aid in the diagnostic process has been a long term goal of research in artificial intelligence. Arguably, it is the most typical application of artificial intelligence.

The different systems that have been developed so-far use a variety of modeling approaches which can be roughly divided into two categories: rule-based approaches with or without uncertainty and probabilistic methods. The rule-based systems can be viewed as computer implementations of the protocols, as described above. They consist of a large data base of rules of the form: $A \rightarrow B$, meaning that ”if condition A is true, then perform action B” or ”if condition A is true, then condition B is also true”. The rules may be deterministic, in which case they are always true, or ’fuzzy’ in which case they are true to a (numerically specified) degree. Examples of such programs are Meditel[5], Quick Medical Reference (QMR)[6], DXplain[7], and Iliad[8].

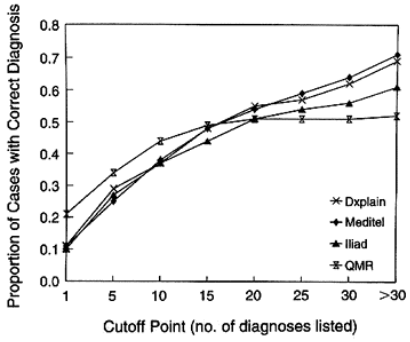


Figure 1: Performances of DSS programs (taken from [9]).

In Berner et al.[9] a detailed study was reported that assesses the performance of these systems. A panel of medical experts collected 110 patient cases, and concensus was reached on the correct diagnosis for each of these patients. For each disease, there typically exists a highly specific test that will unambiguously identify the disease. Therefore, based on such complete data, diagnosis is easy. A more challenging task was defined by removing this defining test from each of the patient cases. The patient cases were presented to the above 4 systems. Each system generated its own ordered list of most likely diseases. In only 10-20 % of the cases, the correct diagnosis appeared on the top of these lists and in approximately 50 % of the cases the correct diagnosis appeared in the top 20 list. Many diagnoses that appeared in the top 20 list were considered irrelevant by the experts. It was concluded that these systems are not suitable for use in clinical practice.

There are two reasons for the poor performance of the rule based systems. One is that the rules that need to be implemented are very complex in the sense that the precondition A above is a conjunction of many factors. If each of these factors can be true or false, there is a combinatoric explosion of conditions that need to be described. It is difficult, if not impossible, to correctly describe all these conditions. The second reason is that evidence is often not deterministic (true or false) but rather probabilistic (likely or unlikely). The above systems provide no principled approach for the combination of such uncertain sources of information.

Probabilistic models

A very different approach is to use probability theory. In this case, one does not model the decision tree directly, but instead models the relations between diagnoses $D_1, \ldots, D_n$ and tests (symptoms, physical examinations, laboratory tests, etc) $T_1, \ldots, T_m$ in one large *probability model*, $P(D_1, \ldots, D_n, T_1, \ldots, T_m)$. In such a model, the differential diagnosis is given by a list of the probabilities $P(D_i|T_C)$, for of each of the diagnoses D_i given the current findings $T_C = T_{C_1}, \ldots, T_{C_k}$. The probability $P(D_i|T_C)$ is computed using the standard axioms of probability theory,

$$P(D_i|T_C) = \frac{P(D_i, T_C)}{P(T_C)} \tag{1}$$

where $P(D_i, T_C)$ is the joint probability of $D_i$and T_C according to the model P, and $P(T_C)$ the probability of the findings T_C. Joint probabilities of a subset of variables (such as D_i, T_C) can be computed by summation over states of the other variables in the system, e.g.,

$$P(D_i, T_C) = \sum_{\{D_{\text{other than } i}, T_{\text{other than } C}\}} P(D_i, D_{\text{other than } i}, T_C, T_{\text{other than } C}) \tag{2}$$

Furthermore, with a probabilistic model one can compute which additional test T_j is expected to provide most information about diagnosis D_i. Suppose the additional information of a test result T_j about diagnosis D_i in the context of the measurements so far is given by $I(D_i, T_j|T_C)$. This information, however, is not accessible, since the state of the diagnosis and of the test to be performed is not known. However, with the probabilistic model one can compute the expected value I_{ij}

$$I_{ij} = \langle I(D_i, T_j|T_C)\rangle = \sum_{\{D_i, T_j\}} P(D_i, T_j|T_C) I(D_i, T_j|T_C) \tag{3}$$

for each test j that has not been measured so-far. The test j that maximizes I_{ij} is the most informative test, since averaged over its possible outcomes, it gives the the most information about D_i. Different information criteria for test selection can be used. For example, one can define a criterium in which the cost of a test is incorporated.

Probabilistic modeling is doable in the framework of Bayesian networks [18] (also known as belief networks, and as graphical models, see [19] for a short introduction). In Bayesian networks the probability model $P(D_1, \ldots, D_n, T_1, \ldots, T_m)$ (which is basically a huge table with about 2^{n+m} entries) is defined by local probabilistic relations of directly related variables. For instance, if one assumes that diagnoses D_i are a priori independent, and that tests T_j are mutually independent *given the state of all diagnosis variables* then one can simplify the model into

$$P(D_1, \ldots, T_m) = \prod_j P(T_j|D_1, \ldots, D_n) \prod_i P(D_i) \tag{4}$$

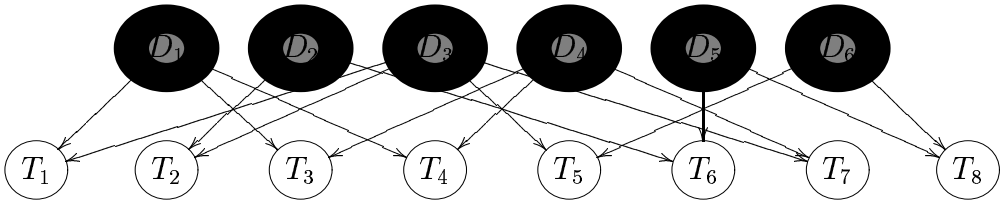


Figure 2: Graphical model corresponding to (5) with K = 2.

The number of parameters is reduced from about 2^{n+m} to $m2^n + n$. If the number of tests, m, is large, this is a considerable saving. If one moreover assumes that each test actually depends on only a few (say K) diagnoses, then even more savings can be obtained. The model reduces to

$$P(D_1, \ldots, T_m) = \prod_j P(T_j | D_{j_1}, \ldots, D_{j_K}) \prod_i P(D_i) \tag{5}$$

The number of parameters is now reduced to $m2^K + n$. This is a significant saving if $K \ll n$. Such models are conveniently expressed as a graph (hence the name graphical models). Each variable in the model corresponds to a node in the graph. Each node and its incoming arrows (parents) correspond to a conditional probability distribution in the factorization of the full joint probability distribution. E.g., in figure 2, node T_1 and its parents D_1 and D_3 correspond to a factor $P(T_1 | D_1, D_3)$ in the joint probability distribution eq. 5

To conclude this introduction, one sees that in rule based systems the diagnostic rules are modeled explicitly, whereas in the probabilistic approach these ‘rules’ are computed ’on the fly’ from the model. For instance, using eq. 3 it could be decided that for a patient for which findings T_C are known, T_i is the most informative next test to rule out diagnosis D_i. This ’rule’ is not explicity stored in the system, but results from Bayesian inference on the graphical model. The advantage of this latter approach is that

1. it is consistent since these ’rules’ are consistently derived from a single model, using the axioms of probabilty, whereas rules from a rule based system may be contradictory

2. the diagnostic advice is always derived in the context of the individual patient condition (probabilities are computed conditioned on the current test results). If one would try to do this in a rule based system, the number of rules would explode.

3. The graphical model representation with probabilities of diseases and conditional probabilities of findings given diseases is more transparent and explicit about the medical knowledge and the presumed mechanisms than a rule based system such as a diagnostic protocol that prescribes which diagnostic steps must

be taken. This generally improves the understanding of what is being modeled and greatly facilitates maintenance (changing probability tables, adding diseases or findings), as well as evaluation by external experts.

Until recently, the drawback of probabilistic modeling was that it was infeasible to construct models with more than a few variables. With the advent of fast computers, and the invention of new inference algorithms for Bayesian networks, much larger models are nowadays doable.

Building large probabilistic networks

The standard procedure is to define a Bayesian network by hand, by specifying a network structure of local interactions and by specifying those probabilities that are needed to define these interactions quantitatively. For medium sized networks (up to 50 - 100 variables), this is doable, for instance with a Bayesian network editor with a graphical user interface , such as BayesBuilder. For larger systems it is more difficult to keep overview, and not to get lost in the spagetthi of relations and interactions. Since our aim is to scale-up the system to 1000's of variables, we automated the generation of probabilistic networks.

We developed a database structure in which medical specialists can enter their knowledge in a way which is familiar for them (see section 3). The database contains information from which the structure of the network can be derived, and the parameters of the network can be learned. In addition, the database contains information about the structure of PROMEDAS's graphical user interface. See section 5). We developed MATLAB subroutines which transforms (a selection) of the database into a complete PROMEDAS application, including interface and engine to do probabilistic inference.

In the following, we sketch how we generate a Bayesian network from the database. The construction of a Bayesian network consists of two parts, a qualitative and a quantitative part. The qualitative part is the determination of the structure of the network. The quantitative part consists of specifying the conditional probability tables (CPTs). First we will explain how we define the structure of the network. We allow three kind of nodes. Prior nodes (risk factors such as occupation, drug use): H_i diagnostic nodes (diseases): D_j, and test nodes (tests, symptoms ...) T_k The structure of network is basically that arrows point from cause to effect. The prior nodes point to diagnostic nodes. Diagnostic nodes can point to other diagnostic nodes and test nodes (see figure 3). The structure of the network (the arrows) are read from the database.

The general structure of the model is of the form eq. 5 (with an extension to include the H_i nodes). The probability of each node is given as a CPT (a conditional probability table $P(T_j|D_k, D_l, D_m)$ specifies the probability of T_j for all combinations of states of D_k, D_l, and D_m). A problem is that the number of parameters to specify each CPT is *exponential* is in number of direct causes (parents). We do not want to restrict the number of parents. Instead, our approach is to parametrize CPTs with a large number

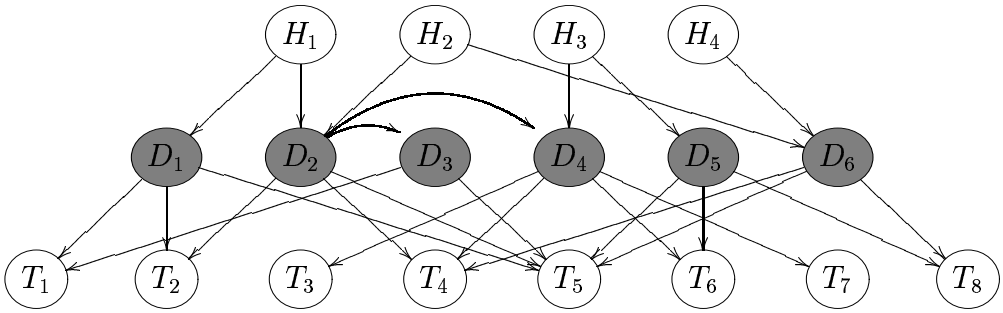


Figure 3: Network structure

of parents as noisy-OR CPTs [18]. The number of parameters to represent noisy-OR CPTs as well as the time required for inference [20] is *linear* in number of parents [1].

Noisy-OR nodes can not represent all possible conditional probability tables and make the following assumptions which are often true in the medical domain: each of the causes of a noisy-OR variable has a probability to trigger the variable to be in a certain state, *regardless* of the states of other causes. If different causes have a probability to trigger the effect to be in the *same* state, then a combination of these causes makes this state *more likely*. If different causes can trigger the effect to be in *different* states, then a combination of these causes makes *any state* (in between) possible. If these assumptions do not apply (which may be indicated in the database), then one can always resort to explicitly modeling of the tables.

After having established the qualitative part of the net, the next part is specifying the entries in the CPTs. These CPTs are also to be set on the basis of expert knowledge. Our approach is to let the domain experts specify the knowledge in a way that is convenient for them (sensitivity of a test, etc.), and then to *learn* the required CPTs. The difficulty is that expert knowledge specified in this way often does not translate directly into network CPTs. This is, for example, the case when conditional probabilities are given in the ‘wrong direction’, from ‘effect’ to ‘cause’ (e.g. one needs $P(T_j|D_i)$ but one only knows $P(D_i|T_j)$, or when only partial information is given (e.g., one needs $P(T_j|D_i, D_k)$ but only $P(T_j|D_i)$ and $P(T_j|D_k)$ is given). The learning consists of a best fit in a high dimensional parameters space with incomplete data [21]. Algorithms [22], for this have been implemented in the MATLAB module.

Inference in large probabilistic networks

A major problem in probabilistic modelling with many variables is the computational complexity involved in typical calculations for inference. For sparsely connected probabilistic networks this problem has been solved during the last decades by the invention of efficient algorithms for exact inference. However, in large, densely connected

[1]The introduction of noisy-OR CPTs is necessary but not sufficient to solve the intractability of inference in a large network consisting of many such CPTs. See next subsection.

models exact inference is intractable. This means that the computation time increases exponentially with the problem size. In such a case, sampling methods, like Markov Chain Monte Carlo (MCMC) may seem a straightforward solution, but may require extreme long computation time to gather sufficiently many samples. An alternative solution is provided by variational methods. These methods do the approximations directly by fitting distributions rather than by gathering statistics from samples.

A very recent development is the application of the Cluster Variation method (CVM) to probabilistic inference [23, 24]. CVM is a method that has been developed in the physics community to approximately compute the properties of the Ising model. The CVM approximates the probability distribution by a number of (overlapping) marginal distributions (clusters). CVM yields exact results in tree structured graphical models. However, it can can also give impressive results for graphs that are not trees. CVM methods are implemented in BayesBuilder. The implementation is still in an experimantal phase. Recently, we have developed a practical improvement upon CVM (submitted for patent). We plan to implement these method in PROMEDAS in near future.

3 The Medical Knowledge

Bayesian networks and their algorithms provide a powerful engine for medical decision making. The first step in the actual development of a diagnostic DSS is the definition and modelling of medical knowledge. The domain is defined, and the knowledge concerning this domain is acquired and finally represented in the network. The acquired knowledge must be stored such that it remains accessible and understandable. In our approach, this is greatly facilitated by separating the knowledge database from the probabilistic model.

Knowledge acquisition and representation

Domain knowledge is acquired from the literature[2] and entered in a knowledge - database (or knowledge base). The knowledge-base contains the following information:

- A diagnostic repertoire, intended to be exhaustive in that context
- A repertoire of findings (sign and symptoms, laboratory results and history), including risk factors and tests that are potentially relevant for that sub-domain
- Specifications of the discretisation and normal values of variables.
- Quantitative relations between variables, such as sensitivities and specificities. Promedas adapts the entries in the conditional probability tables such that the required sensitivities and specificities are correctly reproduced by the model.
- Local prevalences of diseases, as well as more specific pre-test probabilities, specifying special circumstances that influence test results if not taken into account in the model.

[2]For instance, Harrison's principles of internal medicine, Cecil's textbook of medicine, The medical clinics of North America, Up to Date, The Cochrane Library and relevant journal articles retrieved with the help of Medline.

The (conditional) probabilities are determined on the basis of data in the literature or on ”educated guesses” based on local statistics and experience if no data from the literature are available. In addition, information about the organization of the interface is stored.

After acquisition of knowledge, the information in the database is ”compiled” into a model represented by a causal probabilistic network. This has the advantage that only the database is to be maintained. In particular, cut-off points and local prevalences may require adjustments depending on local circumstances. This can be handled in the database. By compilation, the PROMEDAS application will be automatically adapted accordingly. It is also possible to compile networks from part of the database. So there is no need to set up a new database if different versions of PROMEDAS covering different subdomains are to be created. Users will have (partly) access to the content of these databases via help functions and therefore may better interpret the advise given by the system and estimate its merit.

Medical sub-domains

Currently, the database consists of about 752 diseases and 7000 diagnosis-test relations. It covers mainly large parts of endocrinology, diagnosis of aids, and lymphoma diagnosis. Concrete subdomains that have been realized are

- Lipids and vascular diseases. This network consists of 168 diagnoses and a total of 392 findings. The performance of this network will be evaluated in the fall of 2002 by 10-15 specialists in internal medicine The network has been designed by dr. A. van Beek.

- Imuno-phenotyping of malignant lymphomas. This is a relatively small subdomain consisting of 23 tests and 11 diagnoses. It is intended for use in the hematology laboratory as well as by hematologists. The network has been designed by dr. A. Bloem. The network will be evaluated in 2003.

- We have started with the modeling of the remainder of endocrinology. This is a large domain consisting of several 1000 variables. We estimate that we have modeled approximately 40 % of this domain. Much of the modeling of this domain has been done by dr. E. Koning.

4 Evaluation

Finally it is important to evaluate the usefulness of the system. During the development of the system, intermediate evaluation results are used to improve the system. When the system is fully developed, final "real life" evaluation results are used to assess the viability of the general method

During its development the validity and the performance of the system will be tested by the experts in the project team and by clinical experts in endocrinology and hematology at Utrecht University Hospital. The results will be used for a first assessment of the potential benefit of the system, as well as to improve the system performance.

Assessment procedures will be set up to evaluate the diagnostic performance of the system, and to compare it with other systems. A panel of external experts will determine a set of ”gold standard” cases by reaching diagnostic concensus on a sufficiently large set of challenging cases from ”real life”.

Evaluation will include assessment of the performance of the system alone in comparison with the performance of a group of target users with and without using the advise of the system. The performance will also be compared with existing diagnostic DSSs (Gideon, which is currently in use at the UMCU).

PROMEDAS will be implemented step-wise by installation at workstations at the outpatient clinic of interne medicine at UMCU and possibly one or two affiliated community hospitals. Its usefulness in daily clinical practice will be evaluated by closely monitoring and by structured pre-set questionnaires for (target) users.

A version of PROMEDAS for endocrinology will be tested by a team of experts at the department of internal medicine, in the above described protocol. A version of PROMEDAS for lymphoma diagnosis will be tested on a testset of real cases (historical data) at the department immunology (Dr. A. Bloem).

5 Software Development

Software organization

Software plays a crucial role in this project. The software consists of four parts (see figure 4); 1) ACCESS database. A structure is developed in which it is possible to quickly enter and/or adapt large amount of medical knowledge in a way which is familiar to medical experts. 2) Matlab routines to compile (via learning) the content of the database into a Bayesian network and an interface specification. MATLAB is used because it is an advanced and convenient platform for software development. Speed plays only a minor role in this compilation step. 3) The inference engine (C++ JAVA), which is part of our software package BayesBuilder. 4) The PROMEDAS application(C++, JAVA), which is the DSS for the endusers.

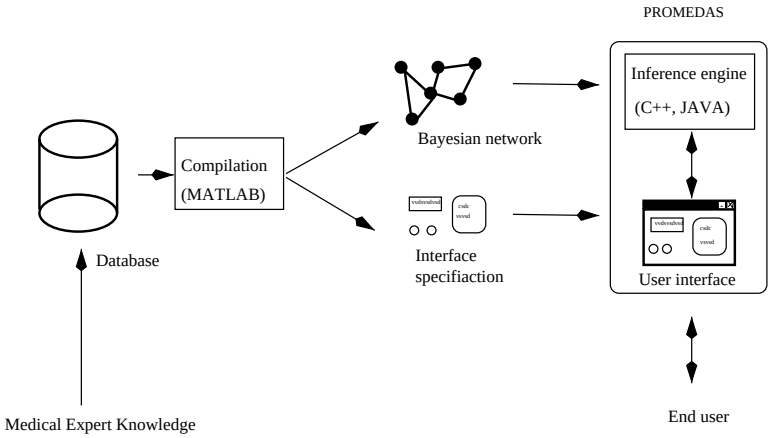


Figure 4: Organization of PROMEDAS development. End user may be connected via an EPD in future.

BayesBuilder

During a previous project we developed a software system, called BayesBuilder, for modelling and inference in Bayesian networks [3]. The most important reasons to develop our own software instead of using commercially available development environments is that it is not possible to integrate and test our approximate inference methods in these systems.

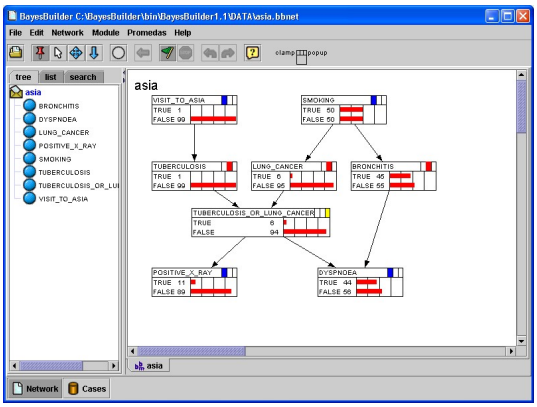


Figure 5: BayesBuilder, loaded with the well-known ASIA network.

The current version of BayesBuilder contains the basic inference algorithms like other (commercial) Bayesian network development environments (HUGIN, Netica). In addition, we have implemented efficient (exact) inference methods for networks with noisy-OR nodes (without these, computation would be intractable). Also approximate methods are implemented, but these are still in an experimental phase.

The BayesBuilder GUI is written in JAVA and the kernel containing the Bayesian inference engine is implemented as a linkable library (API) written in C++. This enables the inference engine to be incorporated in other applications.

It should be noted that the BayesBuilder GUI is only a development tool. This GUI is not the system that is used in clinical practice (although the BayesBuilder API is used for inference.)

In a previous project, we used BayesBuilder to build a PROMEDAS demonstration system. The current version of BayesBuilder is publicly available at the SNN web-site.

By availability of BayesBuilder through the web site, the technology developed in this project is made available to third parties to facilitate future research as well as commercial activities. This objective implies that continued software development is needed in response to user requirements from both inside the project as from outside.

The PROMEDAS interface

The PROMEDAS application (JAVA, C++) is automatically generated from a specification. This specification consists of a Bayesian network and a forms-file. The Bayesian network specifies all the probabilistic relations between variables in the model. The Bayesian network is used for inference in PROMEDAS, by means of the BayesBuilder API. The forms-file specifies the graphical user interface, i.e., which variables are diagnoses, which are tests, and how these are organized in the interface. PROMEDAS’ graphical user interface (GUI) consists basically of two tab panels,

[3] A public version of BayesBuilder can be downloaded from http://www.snn.kun.nl/Research/bayesbuilder/

'Form', and 'Consultant'.

Form This tab panel is for entering clinical data T_C of an individual patient. In this panel, clinical data is organized in categories and subgroups in a similar way as in a conventional paper patient record file. See screenshots (figure 6).

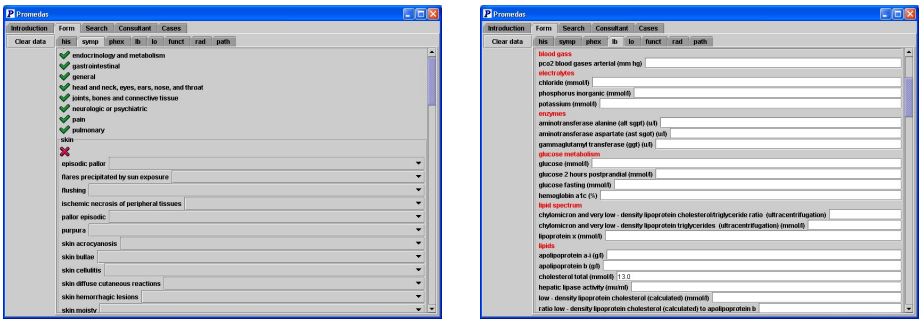


Figure 6: Screenshots from the PROMEDAS data entry panels. Panels with a 'V' button can be expanded into subpanels. Panels with a 'X' button can be colapsed.

Consultant Figure 7 displays the PROMEDAS diagnostic advice panel (Consultant). The advice is computed using the BayesBuidler API. The Consultant panel contains four frames. The lower left frame shows the entered data sofar.

The upper left frame shows the differential diagnosis. It displays the probabilities $P(D_i|T_c)$ of potentially relevant diagnoses D_i (ranked in descending order), given the values T_c of the variables previously entered. Furthermore, it displays the ratio $P(D_i|T_c)/P(D_i)$ between the current probabilities and the prevalences $P(D_i)$ (ranked in descending order). These ratios emphasize the more seldom diagnoses that are related to the patient data, and filters out the irrelevant diagnoses with a high prevalence, but which are irrelevant in the context of the patient data. The user can select a diagnostic category or mechanism for which he/she wants test proposals by a mouse click.

The upper right frame shows the additional test proposals suggested by PROMEDAS. It displays a relative measure (no units) for the information I_{ij} (cf. eqn. (3)) that is to be expected by performing a proposed test T_j *in relation to the selected diagnosis* D_i, (given the values of the variables previously entered). The information is normalized, and displayed in descending order.

The user can select a test proposal T_j for which he/she wants more information by a mouse click. The lower right frame explains why this proposed test is informative for a selected diagnosis.

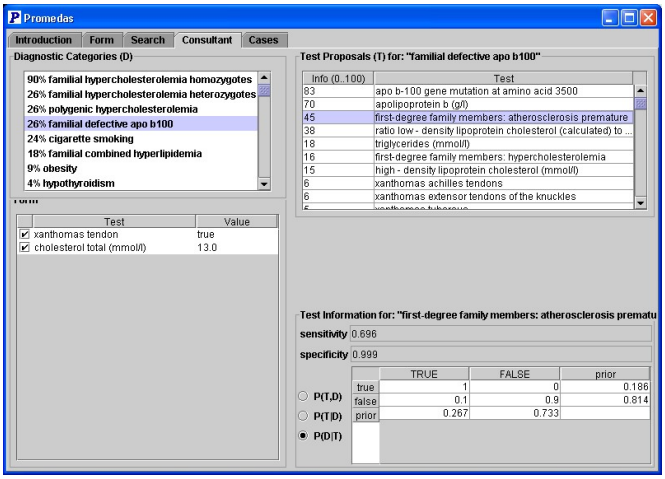


Figure 7: The PROMEDAS consultant page displays the differential diagnosis (upper left); the entered data (lower left); the proposed additional tests with respect to a selected diagnosis (upper right); and the effect of the possible outcomes of the selected test to the selected diagnosis, together with the probability of these possible outcomes (lower right).

6 The PROMEDAS Team

The development of PROMEDAS is a joint project by SNN, Nijmegen and the University Medical Centre Utrecht (UMCU). This project is financially supported in part by the Dutch Technology Foundation (STW).

The PROMEDAS team members are

Dr. H.J. Kappen	SNN	Project management
Dr. W.A.J.J. Wiegerinck	SNN	Artificial Intelligence
E. Akay, MSc	SNN	Software Engineering
M. Nijman, MSc	SNN	Software Engineering
Dr. J.P. Neijt, MD	UMCU	Internal medicine
Dr. A.P. van Beek MD	UMCU	Internal medicine
Dr. E. Koning	UMCU	Internal medicine

WWW:

http://www.mbfys.kun.nl/snn/Research/promedas/

Correspondence Address:

Foundation for Neural Networks
University of Nijmegen
PO Box 9101
6500 HB Nijmegen
The Netherlands
Tel.: +31-(0)24 3614241
Fax.: +31-(0)24 3541435
e-mail:bert@snn.kun.nl
http://www.snn.kun.nl/nijmegen/

References

[1] D. M. Eddy. The challenge. *Journal of American Medical Association*, 263:287–290, 1990.

[2] M. Berg. *Rationalizing Medical Work, Decision-support Techniques and Medical Practices.* MIT press, 1997.

[3] R.J. Lilford, S.G. Pauker, D.A. Braunholz, and J. Chard. Decision analysis and the implementation of research findings. *BMJ*, 317:405–409, 1998.

[4] B. Brigl, P. Ringleb, T. Steiner, P. Knaup, W. Hacke, and R. Haux. An integrated approach for a knowledge-based clinical workstation: Architecture and experience. *Methods of Information in Medicine*, 37:16–25, 1998.

[5] Meditel, Devon, Pa. *Meditel: Computer assisted diagnosis*, 1991.

[6] CAMDAT, Pittsburgh. *QMR (Quick Medical Reference)*, 1992.

[7] Massachussetts General Hospital, Boston. *DXPLAIN*, 1992.

[8] Applied Informatics, Salt Lake City. *ILIAD*, 1992.

[9] E.S. Berner, G.D. Webster, A.A. Shugerman, J.R. Jackson, J. Algina, A.L. Baker, E.V. Ball, C.G. Cobbs, V.W. Dennis, E.P. Frenkel, L.D. Hudson, E.L. Mancall, C.E. Racley, and O.D. Taunton. Performance of four computer-based diagnostic systems. *N-Engl-J-Med.*, 330(25):1792–6, 1994.

[10] ES Berner, JR Jackson, and J Algina. Relationships among performance scores of four diagnostic decision support systems. *J-Am-Med-Inform-Assoc.*, 3(3):208–15, 1996.

[11] D.E. Heckerman, E.J. Horvitz, and B.N. Nathwani. Towards normative expert systems: part I, the Pathfinder project. *Methods of Information in medicine*, 31:90–105, 1992.

[12] D.E. Heckerman and B.N. Nathwani. Towards normative expert systems: part II, probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in medicine*, 31:106–116, 1992.

[13] J.J. Dijkstra. *On the use of computerised decision aids.* PhD thesis, State University of Groningen, 1998.

[14] W. Wiegerinck, W. ter Burg, E. ter Braak, P. van Dam, Y. O, Neijt J., and H Kappen. Inference and advisory system for medical diagnosis. Technical report, SNN, 1997.

[15] WHO. *International Classification of Diseases, 10th revision, Clinical Modification ICD-10*, 1992.

[16] J. Boersma, R.S. Gebel, and H. Lamberts. *International Classification of Primary Care, Short titles (vertaald in het Nederlands) en Nederlandse subtitels.* Nederlands Huisartsen Genootschap, 1993.

[17] WHO. *International Classification of Diseases for Oncology*.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., 1988.

[19] W. Wiegerinck and T. Heskes. *Dealing with the data flow*, chapter Belief networks / Bayesian networks. STT Beweton, 2002.

[20] M. Takinawa and B. D'Ambrosio. Multiplicative factorization of noisy-MAX. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence UAI99*, pages 622–30, 1999.

[21] W. Wiegerinck and T. Heskes. Probability assessment with maximum entropy in Bayesian networks. In A. Goodman and P. Smyth, editors, *Computing Science and Statistics*, volume 33. Interface, 2001. In press.

[22] W. Wiegerinck and Heskes T. IPF for discrete chain factor graphs. In *UAI 2002: Proceedings of the Eightteenth Conference on Uncertainty in Artificial Intelligence*, 2002.

[23] H.J. Kappen and W. Wiegerinck. Novel iteration schemes for the Cluster Variation Method. In T.G. Dieterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, 2002.

[24] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings UAI-2002*, 2002.