Probabilistic knowledge representation

Bert Kappen, Stan Gielen, Tom Heskes,
Wim Wiegerinck, David Barber, Pierre van de Laar
SNN Theory RWI Laboratory
University Nijmegen, the Netherlands

1 Introduction

The aims of this project are to develop novel theory, techniques and implementations for learning and reasoning in a complex dynamic multi-sensory environment. The approach to reasoning and learning is based on the axioms of probability theory and Bayesian statistics. It is argued that such an approach is the most attractive way to design systems for reasoning and learning that are capable of reliable and robust performance in complex real-world environments.

In section 2, we present some novel theoretical results for approximate inference in large graphical models. In section 3, we apply these techniques to medical diagnosis. In section 4 we show how to quantify the uncertainty in the model parameters.

2 Neural Networks and Graphical Models

Neural networks are, in the statistical sense, graphical representations of non-linear functions. Graphical Models, on the other hand, are graphical representations of probability distributions. In the context of uncertain knowledge, Graphical Models therefore, often provide a more natural representation of the problem than neural networks. The core understanding in Graphical Models is that the probability distribution over the variables of interest can be split into subgroups on the basis of an assumed independence structure between variables. A link between two nodes in the graph represents the influence of one variable on the likely state of the other. These models have proved valuable in many sectors of Artificial Intelligence and Machine Learing. Indeed, many traditional models, such as Hidden Markov Models, can be understood within the Graphical Model paradigm. One of the original probabilistic neural network models, the Boltzmann Machine (BM), is readily interpreted as an undirected Graphical Model in this modern framework.

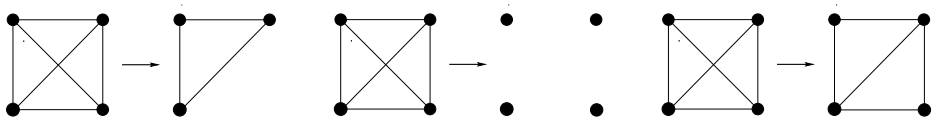
Computing with large Graphical Models uncovers many of the same difficulties underlying the science of complexity - indeed, many of the computations can be shown to be NP-complete. The Boltzmann Machine is an ideal probabilistic model for understanding the difficulties of large computation and also provides a testbed for algorithms. There are close links between the physics of magnetic systems and Boltzmann Machines. Initially, one promising approach for approximating Boltzmann Machines was adapted directly from the physical Mean Field approximation. In contrast with other approximate techniques, such as Monte Carlo methods, Mean Field techniques provide exact bounds on quantities of interest. The central idea is to approximate an intractable Graphi-

cal Model, represented by a discrete distribution $P(S)$, by a simpler distribution $Q(S)$. The parameters of this simpler distribution are found by minimizing the Kullback-Leibler divergence

$$KL = \sum_S \{Q \log Q - Q \log P\} \tag{1}$$

The original application of Mean Field techniques corresponds to assuming a factorized distribution $Q(S) = \prod q_i(S_i)$ which did not provide entirely satisfactory solutions, since the approximation was too limited (fig 1(b)). Very recently, however, there has been a tremendous resurgence in the interest in such "variational" techniques, using more powerful algorithms, whilst retaining the attractive feature of exact bounds on quantities of interest. Graphically, one approach is to uncover tractable subgraphs by removing nodes using mean field methods, fig 1(a).

Part of the recent focus of work within the RWCP has been to extend the variational approximations, increasing their accuracy and applicability to real world problems. This has already succeeded in workable, practical algorithms for both directed and undirected graphs. We have made the observation that, as long as the Kullback-Leibler divergence is calculable, then *any* approximating distribution can be used. We have exploited this to use a class of tractable, "decimatable" Boltzmann Machines as the approximating distributions, fig 1(c). This greatly improves the accuracy of approximation, without greatly increasing the number of variational parameters, in contrast with, for example the mixture approach[1, 2]. An application of our method to approximate the marginal likelihood of the visible units of a toy, directed graph (fig 2), showed the method to be more accurate than existing approximation methods (fig 3). We hope to further extend this method to produce more accurate, bounded variational approximations to many probabilistic models. Indeed, we have also developed procedures that are directly applicable to calculations in statistical physics. We hope to show that this approach will lead to a much clearer solution to some calculational problems in the field of complexity.



(a) Node removal (b) Mean Field (c) Tractable graph

Figure 1: A fully connected 4 node BM is not decimatable. Variational approximations correspond to decimatable subgraphs of varying complexity.

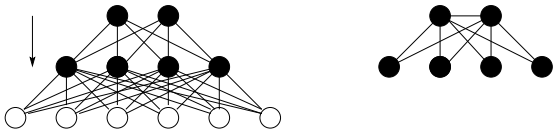
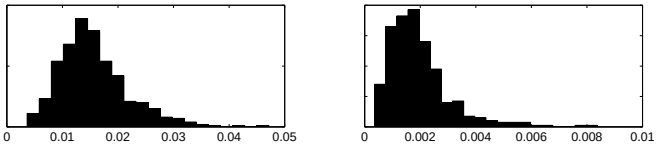


Figure 2: Directed graph toy problem (left). The hidden units (black) are approximated by a BM (right), one of many possible tractable structures.



(a) Mean Field (b) Tractable graph

Figure 3: Histogram of relative error $\ln P_{approx}(V)/\ln P_{exact}(V) - 1$ for 500 random networks - note the different scales. Mean error: (a) 0.0156 (b) 0.0020

3 Medical Diagnosis with Large Probabilistic Networks

The project on Medical Diagnosis with Large Probabilistic Networks is in collaboration with the University Hospital Utrecht. The long term aim of this project is to build a broad and detailed model for internal medicine. The model should be rich and detailed enough to be useful for medical practice. The aim of the model is to perform several types of medical reasoning, such as diagnosis and active decision. In order to make reasoning computationally tractable, newly developed variational methods are used.

In the first phase of the project, we restrict ourself to the medical domain of anemia. With this restriction, modeling and inference is already a non-trivial task, since about 100 diseases and several hundreds of other variables are involved in anemia [3].

3.1 Motivation

Computer-based diagnostic systems can play many roles in decision support and other areas of medical practice. Most systems are designed to produce a differential diagnosis using a set of input findings entered by the user (as opposed to textbooks that tend to do the reverse - taking individual diseases and listing the associated findings). At present several "Diagnostic decision-support tools" are potentially useful such as Meditel, Quick Medical Reference, DXplain, Iliad, and PEM-DXP.

The different systems that have been developed sofar 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 approach can be viewed as an attempt to simplify the probabilistic approach in order to reduce computational complexity. The probabilistic approach has the advantage of mathematical consistency and correctness. In particular belief networks [4])) provide a powerful and conceptual transparent formalism for probabilistic modeling. The progress that has been made during the last decade in exact computation in belief networks makes the argument in favor of rule based approaches less and less persuasive Indeed, most modern approaches for medical diagnosis are based on the probabilistic approach.

The lack of performance of the current systems is therefore not due to the method that is used, but rather due to the level of detail at which the disease areas are modeled. Either the system is based on detailed modeling, but restricted to a small subdomain. Or the system covers a large domain, but at cost of the level of detail at which the disease areas are modeled. The reason for this is that a belief network becomes intractable for exact computation if a large medical area would be modeled in detail. In other words, systems based on exact computation are not able to meet the requirements for general medical practice and one has to resort to approximate methods. In this project we aim to demonstrate the feasibility and the usefulness of this approach.

3.2 Modeling

The problem of building a system for internal medicine can be subdivided into two subproblems: a modeling problem and an inference problem.

The modeling problem is to build a model which includes -up to a satisfactory level- all the necessary knowledge needed for medical reasoning. Our experience has shown, that in general the data available from hospital databases is insufficient to train a detailed model up to a satisfactory level of accuracy. This is also the case for aneamia [1]. Therefore, our approach is to build a belief network on the basis of expert knowledge of physicians.

Building a belief network is only feasible if the number of parents states is limited, or if parametrized nodes are used, such as noisy-OR nodes[4]. However, this restriction is not likely to limit the amount of expert knowledge that can be put in the network, since the expert knowledge of the physicians seem to have exactly the same restrictions.

In addition, medical experts tend to subdivide the medical domain into subdomains, which have only a small overlap. As a result, the final network has a modular structure (cf. fig. 4). Each module represents a subdomain by a reasonably small belief network. The nodes in the subdomains have only a small number of parents. The interconnectivity between the subdomains is also small. There are two types of variables outside the subdomains. One type are variables like 'age' or 'sex', which determine a priori probabilities of diseases. These variabeles are modeled as common ancestors of a large number of subdomains. The other type are variables such as 'headache'. Intuitively, 'headache' is a variable that typically can have its cause in a large number of subdomains. Such variables are modeled as common children of a large number

[1] In some special cases we have found that data of sufficient quality and quantity exist to train useful models on data only

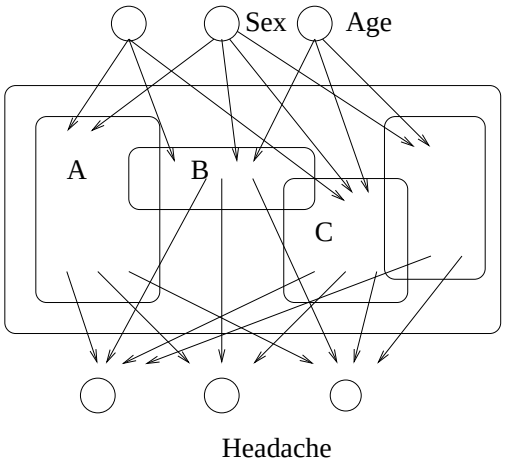


Figure 4: Graphical structure of the modular network. A, B, C ... represent (overlapping) subdomains. Each subdomain is modeled by a number of nodes (not shown in the figure) representing variables that are relevant in that domain. The upper nodes, e.g. 'sex' and 'age' represent common ancestors of nodes in several subdomains. The lower nodes, e.g. 'headache' represent common children of nodes in several subdomains.

of subdomains. Since these nodes have parents in many subdomains, they should be modeled in a parametrized way, such as noisy-OR.

3.3 Approximate Inference

The inference problem is to compute probabilities in the model, given evidence. In the modular network described in the previous section, exact inference on the common children is intractable, since that involves a summation over a large number of parent states. On the other hand, on the upper part of the network - that is the network apart from the common children - exact inference is tractable, thanks its modular structure and sparse connectivity.

In a previous part in this paper, we described how intractable networks can be variationally approximated using tractable graphical structures. These methods can be applied straightforwardly to do approximate inference on the common children. The key step is to use the graphical structure of the upper part of the network to compute bounds of probabilities of the common children. Both upper bounds and lower bounds can be computed in this way, and thus conditional probabilities can be bounded as well.

To conclude, we would like to stress that the variational approximations that we use exploit both the graphical structure of the upper part of the network and the noisy-OR parametrization of the common children, thus enabling very tight bounds.

4 Statistical embedding of learning methods

Neural networks are considered state-of-the-art prediction and classification methods. In almost any benchmark study, they can at least compete with alternative approaches. Despite this apparent success, people sometimes hesitate to use them because of their presumed obscurity: they perform well, but you should not ask why. In regression problems, their inherent nonlinearity makes them more difficult to interpret than simpler alternatives such as linear regression. Similarly, for classification tasks inductive methods as for example decision tree algorithms are much more appealing to our human understanding than neural networks. In the RWCP project, we therefore aim at a better grip on the interpretability and reliability of neural networks. Our methods are based upon a statistical approach: instead of considering a single neural network, we will extract knowledge from ensembles of neural networks.

4.1 Computing error bars

Neural networks are often applied to regression tasks. Error bars then provide a first notion of reliability. In a regression task the goal is to estimate an underlying mathematical function between input and output variables, based on a finite number of data points, possibly corrupted by noise. More specifically, in our database, we have a set of P pairs $\{\vec{x}^\mu, t^\mu\}$, with $\vec{x}$ representing the inputs and t^μ the target, which are all assumed to be generated according to

$$t(\vec{x}) = f(\vec{x}) + \xi(\vec{x}) \, ,$$

where $f(\vec{x})$ is the unknown relationship we are looking for, and where $\xi(\vec{x})$ denotes noise with zero mean. The usual assumption is that this noise is normally distributed with variance independent of $\vec{x}$. The output of a neural network $o(\vec{x})$ can be interpreted as an estimate of the regression $f(\vec{x})$, i.e., of the mean of the target distribution given input variable $\vec{x}$. Sometimes this is all we need to know: a reliable estimate of the regression $f(\vec{x})$. In real-world domains, however, it is important to quantify the accuracy of our statements. Saying that our estimate of the target t^μ is 10 ± 3 can lead to a decision completely different from the one based on 10.16 ± 0.01.

We have developed a method to compute these error bars [5] based on ensembles of neural networks. Different networks are generated by training them on slightly different parts of the available data and by initializing them with different weights. This yields not a single estimate of the regression $o(\vec{x})$, but a whole collection (ensemble) of estimates $o_i(\vec{x})$ where i refers to a particular network.

In [6] we showed that to arrive at the optimal estimate of the regression one should take a weighted average of the individual estimates:

$$\bar{o}(\vec{x}) = \sum_{i=1}^{N} \alpha_i o_i(\vec{x}) \, . \tag{2}$$

where the weighting factors α_i can be computed through a procedure called "balancing". The variance within the ensemble of networks,

$$\sigma^2(\vec{x}) = \frac{1}{1 - \sum_i \alpha_i^2} \sum_{i=1}^{N} \alpha_i [o_i(\vec{x}) - \bar{o}(\vec{x})]^2 \,, \tag{3}$$

immediately provides an estimate of the variance to be used in computing the confidence intervals.

Confidence intervals quantify our confidence in $\bar{o}(\vec{x})$ as an estimate of the true regression $f(\vec{x})$. Usually, we are more interested in prediction intervals. Prediction intervals consider the accuracy with which we can predict the targets $t(\vec{x})$, i.e., deal with the quantity $t(\vec{x}) - \bar{o}(\vec{x})$ instead of $f(\vec{x}) - \bar{o}(\vec{x})$. From,

$$t(\vec{x}) - o(\vec{x}) = [f(\vec{x}) - o(\vec{x})] + \xi(\vec{x}) \,, \tag{4}$$

it is easy to see that a prediction interval necessarily encloses the corresponding confidence interval.

Computing prediction intervals is quite complicated. We have to build a new model to estimate the variance $\chi^2(\vec{x})$ of the noise $\xi(\vec{x})$ inherent to the problem as a function of the inputs $\vec{x}$. For example, $\chi^2(\vec{x})$ may be modelled by a separate neural network. From (4) we then deduce that the variance to be used in the prediction interval should be based on the sum of $\chi^2(\vec{x})$, the variance of the noise inherent to the data, and $\sigma^2(\vec{x})$, the variance corresponding to the confidence interval.

Incorporation of the model uncertainty (the width of the confidence interval) is especially important in regions of input space where there has been hardly any training points. In these regions, the different networks in the ensemble will tend to give quite different results, leading to a relatively large variance $\sigma^2(\vec{x})$ and thus a relatively wide prediction interval (see [5] for an illustration of this effect).

Figure 5 gives an example of confidence intervals and prediction intervals obtained in a real-world problem regarding the prediction of sales figures for department stores.

4.2 Input relevance determination

The inputs of a neural network correspond to explanatory variables, i.e., variables that may have an effect on the output. Afterwards we then can try to quantify the relevance of these explanatory variables: did we really need them or could we as well do without them? Leaving out irrelevant input variables can both lead to better generalization and prediction performance and save resources (no need to collect irrelevant variables). Furthermore, knowing the relevance of input variables increases the user's insight into the problem.

We have applied the above methods to the problem of input relevance determination [7, 8]. Figure 6 illustrates the relevance of the input variables for a problem regarding the sales prediction of deparment stores. We started with an ensemble of networks trained on all input variables. Iteratively we removed the variable that gave the smallest contribution to the percentage of explained

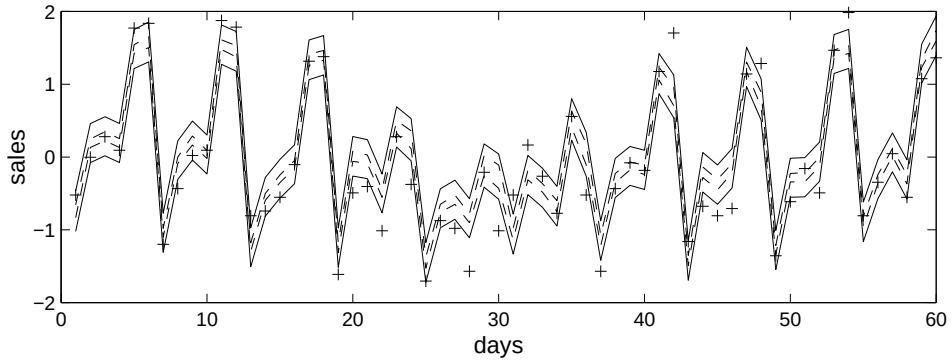


Figure 5: Standard confidence (dashed lines) and prediction (solid lines) intervals for the expected sales figures of a department store in 10 consecutive weeks. Plusses indicate observed sales figures.

variance, averaged over all outputs. In this way we obtained Figure 6, which from right to left, gives the order in which the groups of input variables were eliminated. The remaining group of input variables (not shown) corresponds to the day of the week, which explains about 80% of the variance in the data. The hatched areas show the contributions of each group of input variables, which together with the filled areas yield the cumulative sums.

[1] W. Wiegerinck and D. Barber. Mean Field Theory based on Belief Networks for Approximate Inference. In *ICANN 98*, pages 499–504, 1998.

[2] D. Barber and W Wiegerinck. Tractable Undirected Approximations for Graphical Models. In *ICANN'98: International Conference on Artificial Neural Networks, Skövde*, pages 93–98, 1998. R-98-001.

[3] W. Wiegerinck and H.J. Kappen. Lab-test selection in diagnosis of anaemia. In *Proceedings RWC*, pages 83–88, Tokyo, Japan, 1997. SNN–96–031,TR–96-001.

[4] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, California, 1988.

[5] T. Heskes. Practical confidence and prediction intervals. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 176–182, Cambridge, 1997. MIT Press. SNN–96–038, F–96–034.

[6] T. Heskes. Balancing between bagging and bumping. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 466–472, Cambridge, 1997. MIT Press. SNN–96–039, F–96–032.

[7] P. van de Laar, T. Heskes, and S. Gielen. Partial retraining: a new approach to input relevance determination. *International Journal of Neural Systems*, 9:75–85, 1999. F96-129.

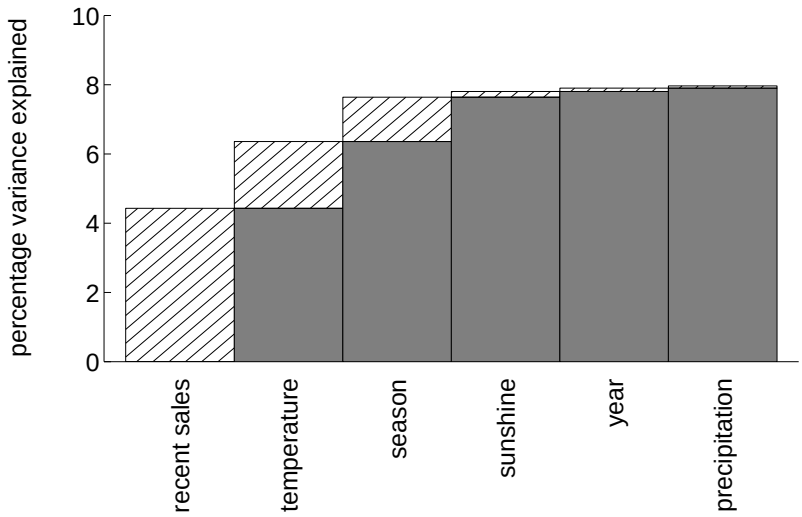


Figure 6: Additional percentages of variances explained by different groups of variables as obtained using partial retraining. The day of the week explains about 80% of the variance in the data (not shown). This is an illustration on real-world data regarding the sales prediction of department stores.

[8] Piërre van de Laar, Stan Gielen, and Tom Heskes. Input selection with partial retraining. In Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud, editors, *Artificial Neural Networks - ICANN'97*, volume 1327 of *Lecture Notes in Computer Science*, pages 469–474, Berlin, 1997. Springer. R-97-023, SNN-97-003.