MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL AND COMPUTATIONAL LEARNING
DEPARTMENT OF BRAIN AND COGNITIVE SCIENCES

A.I. Memo No. 1571 March, 1996
C.B.C.L. Memo No. 136

Computing upper and lower bounds on likelihoods in intractable networks

Tommi S. Jaakkola[†] **and Michael I. Jordan**[‡]
{tommi,jordan}@psyche.mit.edu

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.

Abstract

We present techniques for computing upper and lower bounds on the likelihoods of partial instantiations of variables in sigmoid and noisy-OR networks. The bounds determine confidence intervals for the desired likelihoods and become useful when the size of the network (or clique size) precludes exact computations. We illustrate the tightness of the obtained bounds by numerical experiments.

[†]http://web.mit.edu/~tommi/
[‡]http://www.ai.mit.edu/projects/cbcl/web-pis/jordan/homepage.html

This report describes research done at the Dept. of Brain and Cognitive Sciences, the Center for Biological and Computational Learning, and the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for CBCL is provided in part by a grant from the NSF (ASC–9217041). Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Dept. of Defense. The authors were supported by a grant from the McDonnell-Pew Foundation, by a grant from Siemens Corporation, by a grant ¿from Daimler-Benz Systems Technology Research, and by a grant from the Office of Naval Research. Michael I. Jordan is a NSF Presidential Young Investigator.

1 Introduction

A graphical model provides an explicit representation of qualitative dependencies among the variables associated with the nodes of the graph (Pearl, 1988). Assigning values (potentials or probability tables) to the links connecting the variables in these models enables numerical (or quantitative) computation of beliefs about the values of the variables on the basis of acquired evidence. The computations involved, i.e., propagation of beliefs, can be handled by now standard exact methods (Lauritzen & Spiegelhalter, 1988, Jensen et al. 1990). Junction trees serve as representational platforms for these exact probabilistic calculations and are constructed from directed graphical representations via moralization and triangulation. Although powerful in utilizing the structure of the underlying networks, junction trees may, in some cases, contain cliques that are prohibitively large. We focus in this paper on methods for dealing with such large (sub)structures.

Large clique sizes lead not only to long execution times but also involve exponentially many parameters that must be assessed or learned. The latter issue is generally addressed via parsimonious representations such as the logistic sigmoid (Neal, 1992) or the noisy-OR function (Pearl, 1988). We consider both of these representations in the current paper. We stay within a directed framework and thereby retain the compactness of these representations throughout our inference and estimation algorithms.

As an alternative to sampling methods in intractable networks we develop principled approximations by computing upper and lower bounds on likelihoods of partial instantiatiations of variables. Such bounds can be combined to give rise to confidence intervals for the desired likelihoods (e.g. for node marginals). Although the problem of finding confidence intervals to a predescribed accuracy is NP-hard (Dagum and Luby, 1993), bounds that can be computed efficiently may nevertheless yield confidence intervals that are accurate enough to be useful in practice.

Saul et al. (1996) derived a rigorous *lower* bound for sigmoid belief networks and we complete the picture here by developing the missing *upper* bounds for sigmoid networks. We also develop both upper and lower bounds for noisy-OR networks. While the lower bounds we obtain are applicable to generic network structures, the upper bounds are currently restricted to two-level networks. Although a serious restriction, there are nonetheless many potential applications for such upper bounds, including the probabilistic reformulation of the QMR knowledge base (Shwe et al., 1991). We emphasize finally that our focus in this paper is on techniques of bounding rather than on all–encompassing inference algorithms; tailoring the bounds for specific problems or merging them with exact methods may yield a considerable advantage.

The paper is structured as follows. Section 2 introduces sigmoid belief networks, develops the techniques for upper and lower bounds, and gives preliminary numerical analysis of the accuracy of the bounds. Section 3 is devoted to the analogous results for noisy-OR networks. In section 4 we summarize the results and describe some future work.

2 Sigmoid belief networks

Sigmoid belief networks are (directed) probabilistic networks defined over binary variables $S_1, \ldots, S_n$. The joint distribution for the variables has the usual decompositional structure:

$$P(S_1, \ldots, S_n|\theta) = \prod_i P(S_i|\mathrm{pa}[i], \theta) \qquad (1)$$

The conditional probabilities, however, take a particular form given by

$$\begin{aligned} P(S_i|\mathrm{pa}[i], \theta) &= \\ &= g\big(\textstyle\sum_{j\in \mathrm{pa}[i]} \theta_{ij} S_j\big)^{S_i}\big[1 - g\big(\textstyle\sum_{j\in \mathrm{pa}[i]} \theta_{ij} S_j\big)\big]^{1-S_i} \qquad (2)\\ &= g\big((2S_i - 1)\textstyle\sum_{j\in \mathrm{pa}[i]} \theta_{ij} S_j\big) \qquad (3)\end{aligned}$$

where $g(x) = 1/(1+\exp(-x))$ is the logistic function (also called a "sigmoid" function based on its graphical shape; see Figure 3). The parameters specifying these conditional probabilities are the real valued "weights" θ_{ij}. We note that the choice of this dependency model is not arbitrary but is rooted in logistic regression in statistics (McCullagh & Nelder, 1983). Furthermore, this form of dependency corresponds to the assumption that the odds from each parent of a node combine multiplicatively; the weights θ_{ij} in this interpretation bear a relation to log-odds.

In the remainder of this section we present techniques for computing upper and lower bounds on the likelihood of any instantiation of variables in sigmoid networks. We note that the upper bounds are restricted to two-level (bipartite) networks while the lower bounds are valid for arbitrary network structures.

2.1 Upper bound for sigmoid network

We restrict our attention to two-level directed architectures. The joint probability for this class of models can be written as

$$\begin{aligned} P(S_1, \ldots, S_n|\theta) &= \prod_{i\in L_1} g\big((2S_i - 1)\textstyle\sum_{j\in \mathrm{pa}[i]} \theta_{ij} S_j\big) \\ &\quad \times \prod_{j\in L_2} P(S_j|\theta_j) \qquad (4)\end{aligned}$$

where L_1 and L_2 signify the two layers of a bipartite graph with connections from L_2 to L_1.

To compute the likelihood of an instantiation of variables in these networks, we note that (i) any *instantiated* variables in layer L_2 only reduce the complexity of the calculations, and (ii) the form of the architecture makes any *uninstantiated* variables in L_1, or the "receiving" layer, inconsequential. We will thus adopt a simplifying notation in which the evidence consists of all and only the variables in L_1. Thus, the goal is to compute

$$P(\{S_i\}_{i\in L_1}|\theta) = \sum_{\{S_j\}_{j\in L_2}} P(S_1, \ldots, S_n|\theta) \qquad (5)$$

Given our assumption that computing the likelihood is intractable, we seek an upper bound instead. Let us briefly outline our strategy. The goal is to simplify the joint distribution such that the marginalization across L_2 can be accomplished efficiently, while maintaining at all times a rigorous upper bound on the likelihood. Our approach is to introduce additional parameters into the problem (known as "variational parameters") such that the resulting joint probability distribution factorizes over the uninstantiated variables. Thus we first find a "variational" form for the joint distribution. Although the variational forms are exact they can be turned into upper bounds by not carrying out the minimizations involved and instead fixing the variational parameters. As we will see below this type of variational bound can be obtained by combining variational representations for each sigmoid function in our probability model. We note finally that the variational parameters that are kept fixed during the likelihood calculation can be employed afterwards to optimize the likelihood bound. In essence, this amounts to exchanging the order of the summation over the uninstantiated variables and the variational minimization.

To derive the upper bound we first make use of the following variational transformation of the sigmoid function (see appendix A):

$$g(x) = \frac{1}{1+e^{-x}} = \min_{\xi\in[0,1]} e^{\xi x - H(\xi)} \tag{6}$$

where $H(\cdot)$ is the binary entropy function. Inserting this transformation into the probability model we find

$$\begin{aligned}
&P(S_1,\ldots,S_n|\theta) = \\
&= \prod_{i\in L_1} \min_{\xi_i}\left\{ e^{-H(\xi_i)+\xi_i(2S_i-1)\sum_j \theta_{ij}S_j} \right\} \prod_{j\in L_2} P(S_j|\theta_j) \\
&= \min_{\xi}\left\{ e^{-\sum_{i\in L_1} H(\xi_i)} \times \right. \\
&\quad \left. \times \prod_{j\in L2} \left[e^{\sum_{i\in L_1}\xi_i(2S_i-1)\theta_{ij}} \right]^{S_j} P(S_j|\theta_j) \right\} \qquad (7) \\
&\stackrel{def}{=} \min_{\xi}\{\, \tilde{P}(S_1,\ldots,S_n|\theta,\xi)\,\} \qquad (8)
\end{aligned}$$

where we have pulled the minimizations outside and combined the terms that depend on each of the uninstantiated variables S_j in L_2. This reorganization shows that $\tilde{P}(S_1,\ldots,S_n|\theta,\xi)$ (defined implicitly) factorizes over $\{S_j\}_{j\in L2}$. A simple upper bound on the likelihood is thus obtained in closed form by exchanging the order of the summation and the minimization:

$$\begin{aligned}
P(\{S_i\}_{i\in L_1}|\theta) &= \sum_{\{S_j\}_{j\in L_2}} P(S_1,\ldots,S_n|\theta) \qquad (9)\\
&= \sum_{\{S_j\}_{j\in L_2}} \min_{\xi}\{\, \tilde{P}(S_1,\ldots,S_n|\theta,\xi)\,\} \qquad (10)\\
&\le \min_{\xi} \sum_{\{S_j\}_{j\in L_2}} \tilde{P}(S_1,\ldots,S_n|\theta,\xi) \qquad (11)\\
&= \min_{\xi}\left\{ e^{-\sum_{i\in L_1} H(\xi_i)} \times \right. \\
&\quad \left. \prod_{j\in L_2}\left(P(S_j{=}1|\theta_j) e^{\sum_{i\in L_1}\xi_i(2S_i-1)\theta_{ij}} + P(S_j{=}0|\theta_j) \right) \right\} \qquad (12)
\end{aligned}$$

We state here a few facts about the bound (mostly without proof): (i) The bound can never be greater than one since one is always achieved by setting all ξ to zero, (ii) the bound becomes exact in the limit of small parameter values, and (iii) for fixed prior probabilities $P(S_j|\theta_j)$ the bound has a lower limit and therefore cannot follow closely the true likelihood for very improbable events.

To simplify the minimization with respect to ξ we can work on a log scale and make use of the following Legendre transformation:

$$\log x = \min_{\lambda}\{\lambda x - \log\lambda - 1\} \tag{13}$$

As a result we get

$$\begin{aligned}
\log P(\{S_i\}_{i\in L_1}|\theta) &\le -\sum_{i\in L_1} H(\xi_i) \\
&+ \sum_{j\in L_2} \lambda_j \left(P(S_j{=}1|\theta_j) e^{\sum_{i\in L_1}\xi_i(2S_i-1)\theta_{ij}} + P(S_j{=}0|\theta_j) \right) \\
&+ \sum_{j\in L_2} [-\log\lambda_j - 1] \qquad (14)
\end{aligned}$$

where we have ceased to indicate explicitly that the bound will be minimized over the adjustable parameters. This new form of the bound has the advantage that the minimization with respect to each parameter (ξ or λ) is reduced to convex optimization[1] and can be done by any standard method (e.g. Newton-Raphson). Note that the accuracy of the bound is not compromised by the additional Legendre transformation. Its effect is merely to simplify the expressions for optimization.

2.2 Generic lower bound for sigmoid network

Methods for finding approximate lower bounds on likelihoods were first presented by Dayan, et al. (1995) and Hinton, et al. (1995) in the context of a layered network known as the "Helmholtz machine." Saul, et al. (1996) subsequently showed how such bounds could be made rigorous (by appeal to mean field theory) in the case of generic sigmoid networks. Unlike the method for obtaining upper bounds presented in the previous section, the lower bound methodology poses no constraints on the network structure. We briefly introduce the idea here (for details see Saul, et al.).

Let us denote the set of instantiated variables by $\{S_i\}_{i\in L}$. A lower bound on the (log) likelihood can be found directly via Jensen's inequality:

$$\begin{aligned}
&\log P(\{S_i\}_{i\in L}|\theta) = \\
&\quad = \log \sum_{\{S\}_{i\notin L}} P(S_1,\ldots,S_n|\theta)
\end{aligned}$$

[1] The convexity with respect to each ξ follows from the convexity of e^x and the positivity of the multiplying coefficients λ.

$$= \log \sum_{\{S_i\}_{i \notin L}} Q(\{S\}) \frac{P(S_1, \ldots, S_n|\theta)}{Q(\{S\})}$$
$$\geq \sum_{\{S\}_{i \notin L}} Q(\{S\}) \log \frac{P(S_1, \ldots, S_n|\theta)}{Q(\{S\})} \quad (15)$$

which holds for any distribution Q over the uninstantiated variables $\{S\}$. The bound becomes exact if $Q(\{S\})$ can represent the true posterior distribution $P(\{S\} \mid \{S_i\}_{i\in L}, \theta)$. For other choices of Q the accuracy of the bound is characterized by the Kullback-Leibler distance between Q and the posterior. As we are assuming that computing the likelihood exactly is intractable the idea is to find a distribution Q that can be computed efficiently. The simplest of such distributions is the completely factorized ("mean field") distribution:

$$Q(\{S\}) = \prod_j \mu_j^{S_j} (1 - \mu_j)^{1-S_j} \quad (16)$$

Inserting this distribution into the lower bound (eq. (15)) we can, in principle, carry out the summation[2] and get an expression for the lower bound. Consequently, the adjustable parameters μ_j can be modified to make the bound tighter.

For later utility we rewrite the lower bound in eq. (15) as

$$\log P(\{S_i\}_{i\in L}|\theta)$$
$$\geq E_Q\{ \log P(S_1, \ldots, S_n|\theta) \} + H_Q \quad (17)$$
$$= \sum_i E_Q\{ \log P(S_i|\mathrm{pa}[i], \theta) \} + H_Q \quad (18)$$

where H_Q is the entropy of the Q distribution and $E_Q\{\cdot\}$ is the expectation with respect to Q. We note finally that developing the bound further is highly dependent on the type of the network – whether sigmoid, noisy-OR, or other[3].

2.3 Numerical experiments for sigmoid network

In testing the accuracy of the developed bounds we used $8 \rightarrow 8$ networks (complete bipartite graphs), where the network size was chosen to be small enough to allow exact computation of the true likelihood for purposes of comparison. The method of testing was as follows. The parameters for the $8 \rightarrow 8$ networks were drawn from a Gaussian prior distribution and a sample from the resulting joint distribution of the variables was generated. The variables in the "receiving" layer of the bipartite graph were instantiated according to the sample. The true likelihood as well as the upper and lower bounds were computed for the instantiation. The resulting bounds were assessed by employing the relative error in log-likelihood, i.e. $(\log P_{\mathrm{Bound}} / \log P - 1)$, as a measure of accuracy.

More precisely, the prior distribution over the parameters was taken to be

$$P(\theta) = \prod_i \prod_{j \in \mathrm{pa}[i]} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\theta_{ij}^2} \quad (19)$$

where the overall variance σ^2 allows us to vary the degree to which the resulting parameters make the two layers of the network dependent. For small values of σ^2 the layers are almost independent whereas larger values make them strongly interdependent. To make the situation worse for the bounds[4] we enhanced the coupling of the layers by setting $P(S_j|\theta_j) = 1/2$ for all the uninstantiated variables, i.e., making them maximally variable.

In order to make the accuracy of the bounds commensurate with those for the noisy-OR networks reported below, we summarize the results via a measure of interlayer dependence. This dependence was measured by

$$\sigma_{std} = \sqrt{\mathrm{Var}\{P(S_i|\mathrm{pa}[i])\}} \quad (20)$$

that is, the variability of the likelihood of the instantiation due to different configurations of the uninstantiated variables. Figure 1 illustrates the accuracy of the bounds as measured by the relative log-likelihood as a function of σ_{std}[5]. In terms of probabilities, a relative error of ϵ translates into a $P^{1+\epsilon}$ approximation of the true likelihood P. Note that the relative error is always positive for the upper bound and negative for the lower bound. The figure indicates that the bounds are accurate enough to be useful. In addition, we see that the the upper bound deteriorates faster with increasingly coupled layers.

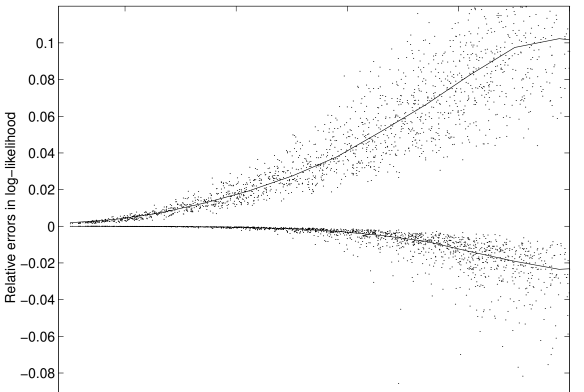

Figure 1: Accuracy of the bounds for sigmoid networks. The solid lines are the median relative errors in log-likelihood as a function of σ_{std}. The upper and lower curves correspond to the upper and lower bounds respectively.

[2] The summation even in case of simple factorized distributions can be non-trivial to perform; see Saul, et al.

[3] For a derivation of lower bounds for networks with cumulants replacing the sigmoid function see Jaakkola et al. (1996).

[4] Both the upper and lower bounds are exact in the limit of lightly coupled layers.

[5] Note that the maximum value for σ_{std} is $1/2$.

3 Noisy-OR networks

Noisy-OR networks – like sigmoid networks – can be represented by DAGs and are written as a product form for the joint distribution:

$$P(S_1,\ldots,S_n|\theta) = \prod_i P(S_i|\text{pa}[i],\theta) \tag{21}$$

Unlike sigmoid networks, however, the conditional probabilities for a noisy-OR network are defined as

$$\begin{aligned} P(S_i|\text{pa}[i],\theta) &= \left(1-\prod_{j\in\text{pa}[i]}(1-q_{ij})^{S_j}\right)^{S_i} \\ &\times \left(\prod_{j\in\text{pa}[i]}(1-q_{ij})^{S_j}\right)^{1-S_i} \end{aligned} \tag{22}$$

where, for example, the parameter q_{ij} corresponds to the probability that the j^{th} parent of i alone can turn S_i on. A constant "leak" or "bias" can be included by introducing a dummy (parent) variable whose value is always fixed to one.

In the following two sections we develop methods for computing upper and lower bounds on the likelihood of any instantiation of variables in the noisy-OR network. Similarly to the case of sigmoid networks the upper bound is applicable to a restricted class of networks while the lower bound remains generic. For clarity of the forthcoming derivations we introduce the notation:

$$\begin{aligned} P(S_i=0|\text{pa}[i],\theta) &= \prod_{j\in\text{pa}[i]}(1-q_{ij})^{S_j} \\ &= e^{-\sum_{j\in\text{pa}[i]}\theta_{ij}S_j} \end{aligned} \tag{23}$$

with $\theta_{ij} = -\log(1-q_{ij}) \geq 0$.

3.1 Upper bound for noisy-OR network

The motivation and, in broad outline, the upper bound derivation itself can be carried over from the sigmoid setting to the noisy-OR case.

Consider a two-level or bipartite network with $\{S_i\}_{i\in L_1}$ and $\{S_i\}_{i\in L_2}$ (where $L_2 \to L_1$) denoting the two sets of variables. As before we adopt a simplifying notation in which an instantiation consists of values for all the variables in the layer L_1. To compute the likelihood of such an instantiation we need to sum the noisy-OR joint distribution,

$$\begin{aligned} &P(S_1,\ldots,S_n|\theta) = \\ &= \prod_{i\in L_1}(1-e^{-\sum_j\theta_{ij}S_j})^{S_i}e^{-(1-S_i)\sum_j\theta_{ij}S_j} \\ &\quad\times\prod_{j\in L_2}P(S_j|\theta_j) \end{aligned} \tag{24}$$

over the uninstantiated variables in L_2. We note that the complexity of performing this calculation exactly increases exponentially with the number of variables that are instantiated to one; importantly, and unlike in the sigmoid case, the complexity does not vary exponentially with the number of uninstantiated variables. Nevertheless, we focus on the case where the exact method of obtaining the likelihood is infeasible.

To find an upper bound in the noisy-OR setting we use the following variational transformation (for a derivation and discussion see appendix B)

$$1-e^{-x} = \min_{\xi\geq 0} e^{\xi x - F(\xi)} \tag{25}$$

where $F(\xi) = -\xi\log\xi + (\xi+1)\log(\xi+1)$. By inserting this transformation into the joint distribution we obtain:

$$\begin{aligned} &P(S_1,\ldots,S_n|\theta) = \\ &= \prod_{i\in L_1}\min_{\xi_i}\left\{e^{S_i[\sum_j\theta_{ij}S_j - F(\xi_i)]}\right\}e^{-(1-S_i)\sum_j\theta_{ij}S_j} \\ &\quad\times\prod_{j\in L_2}P(S_j|\theta_j) && (26) \\ &= \min_\xi\left\{e^{-\sum_{i\in L_1}S_iF(\xi_i)}\times\right. \\ &\quad\left.\prod_{j\in L_2}\left[e^{\sum_{i\in L_1}(S_i\xi_i+S_i-1)\theta_{ij}}\right]^{S_j}P(S_j|\theta_j)\right\} && (27) \\ &\stackrel{def}{=} \min_\xi\{\ \tilde{P}(S_1,\ldots,S_n|\theta,\xi)\ \} && (28) \end{aligned}$$

where we have regrouped terms by rewriting the product over $i \in L_1$ as a sum in the exponent and collecting the terms depending on the uninstantiated variables S_j. We can see that the implicitly defined (and unnormalized) $\tilde{P}(S_1,\ldots,S_n|\theta,\xi)$ factorizes over S_j. This factorial property allows us to find a closed form upper bound on the likelihood:

$$\begin{aligned} &P(\{S_i\}_{i\in L_1}|\theta) = \\ &= \sum_{\{S_j\}_{j\in L_2}}P(S_1,\ldots,S_n|\theta) \\ &= \sum_{\{S_j\}_{j\in L_2}}\min_\xi\tilde{P}(S_1,\ldots,S_n|\theta,\xi) && (29) \\ &\leq \min_\xi\sum_{\{S_j\}_{j\in L_2}}\tilde{P}(S_1,\ldots,S_n|\theta,\xi) && (30) \end{aligned}$$

where the last summation can now be performed exactly to yield:

$$\begin{aligned} &P(\{S_i\}_{i\in L_1}|\theta) \leq \\ &\min_\xi\left\{e^{-\sum_{i\in L_1}S_iF(\xi_i)}\times\right. \\ &\left.\prod_{j\in L_2}\left(P(S_j{=}1|\theta_j)e^{\sum_{i\in L_1}(S_i\xi_i+S_i-1)\theta_{ij}}+P(S_j{=}0|\theta_j)\right)\right\} \end{aligned} \tag{31}$$

This bound (i) always stays below (or equal to) one as it is less than or equal to one whenever all ξ are set to zero, and (ii) is exact when all S_i in L_1 are zero or in the limit of vanishing parameters θ_{ij}.

As in the sigmoid case we may simplify the minimization process by considering $\log P(\{S_i\}_{i\in L_1}|\theta)$ and introducing a Legendre transformation for $\log(\cdot)$. This yields:

$$\log P(\{S_i\}_{i\in L_1}|\theta) \leq \sum_{i\in L_1} S_i F(\xi_i)$$
$$+ \sum_{j\in L_2} \lambda_j \left(P(S_j=1|\theta_j) e^{\sum_{i\in L_1}(S_i\xi_i + S_i - 1)\theta_{ij}} + P(S_j=0|\theta_j) \right)$$
$$+ \sum_{j\in L_2} [-\log\lambda_j - 1] \quad (32)$$

where we have dropped the explicit reference to minimization. The gain again is the convexity of the bound with respect to any of the ξ or λ variables.

3.2 Generic lower bound for noisy-OR network

The earlier work on lower bounds by Saul, et al. was restricted to sigmoid networks; we extend that work here by deriving a lower bound for generic noisy-OR networks. We refer to section 2.2 for the framework and commence from the noisy-OR counterpart of eq. (18). Thus,

$$\log P(\{S_i\}_{i\in L}|\theta)$$
$$\geq \sum_i E_Q\{ \log P(S_i|\mathrm{pa}[i],\theta) \} + H_Q \quad (33)$$
$$= \sum_i E_Q\{ S_i \log(1 - e^{-\sum_j \theta_{ij} S_j}) \}$$
$$+ \sum_i E_Q\{ -(1-S_i)\sum_j \theta_{ij} S_j \} + H_Q \quad (34)$$

which is obtained by writing explicitly the form of the conditional probabilities for noisy-OR networks. While the second expectation in eq. (34) simply corresponds to replacing the binary variables S_i with their means μ_i (since Q is factorized), the first expectation lacks a closed form expression. To compute this expectation efficiently we make use of the following expansion:

$$1 - e^{-x} = \prod_{k=0}^{\infty} g(2^k x) \quad (35)$$

where $g(\cdot)$ is the sigmoid function (see appendix C). This expansion converges exponentially fast and thus only a few terms need to be included in the product for good accuracy. By carrying out this expansion in the bound above and explicitly using the form of the sigmoid function we get

$$\log P(\{S_i\}_{i\in L}|\theta)$$
$$\geq \sum_i \sum_k E_Q\{ -S_i \log(1 + e^{-2^k \sum_j \theta_{ij} S_j}) \}$$
$$- \sum_i (1-\mu_i) \sum_j \theta_{ij}\mu_j + H_Q \quad (36)$$

Now, as the parameters θ_{ij} are non-negative,

$$e^{-2^k \sum_j \theta_{ij} S_j} \in [0,1]$$

and we may use the smooth convexity properties of $-\log(1+x)$ (for $x \in [0,1]$) to bring the expectations in eq. (36) inside the log. This results in

$$\log P(\{S_i\}_{i\in L}|\theta)$$
$$\geq \sum_{ik} -\mu_i \log\left[1 + \prod_j (\mu_j e^{-2^k\theta_{ij}} + 1 - \mu_j)\right]$$
$$- \sum_i (1-\mu_i)\sum_j \theta_{ij}\mu_j + H_Q \quad (37)$$

A more sophisticated and accurate way of computing the expectations in eq. (36) is discussed in appendix D.

3.3 Numerical experiments for noisy-OR network

The method of testing used here was, for the most part, identical to the one presented earlier for sigmoid networks (section 2.3). The only difference was that the prior distribution over the parameters defining the conditional probabilities was chosen to be a Dirichlet instead of a Gaussian:

$$q_{ij} \sim n(1-q_{ij})^{n-1} \quad (38)$$

(recall that $P(S_i = 0|\mathrm{pa}[i],\theta) = \prod_{j\in\mathrm{pa}[i]}(1-q_{ij})^{S_j}$). For large n, q stays small (or $1-q \approx 1$) and the layers of the bipartite network are only weakly connected; smaller values of n, on the other hand, make the layers strongly dependent. We thus used n to vary (on average) the interdependence beweeen the two layers. To facilitate comparisons with the bounds derived for sigmoid networks we used σ_{std} (see eq. (20)) as a measure of dependence between the layers.

Figure 2 illustrates the accuracy of the computed bounds as a function of σ_{std}[6]. The samples with zero relative error are from the upper bound in cases where all the instantiated variables are zero since the bound becomes exact whenever this happens. The lower bound is slightly worse than the one for sigmoid networks most likely due to the symmetry and smoother nature of the sigmoid function. As with the sigmoid networks the upper bound becomes less accurate more quickly.

4 Discussion and future work

Applying probabilistic methods to real world inference problems can lead to the emergence of cliques that are prohibitively large for exact algorithms (for example, in medical diagnosis). We focused on dealing with such large (sub)structures in the context of sigmoid belief networks and noisy-OR networks. For these networks we developed techniques for computing upper and lower bounds on the likelihoods of partial instantiations of variables. The bounds serve as an alternative to sampling methods in the presence of intractable structures. They can define confidence intervals for the likelihoods and can be used to improve the accuracy of decision making in intractable networks.

[6] The slight unevenness of the samples are due to the nonlinear relationship between the Dirichlet parameter n and σ_{std}.

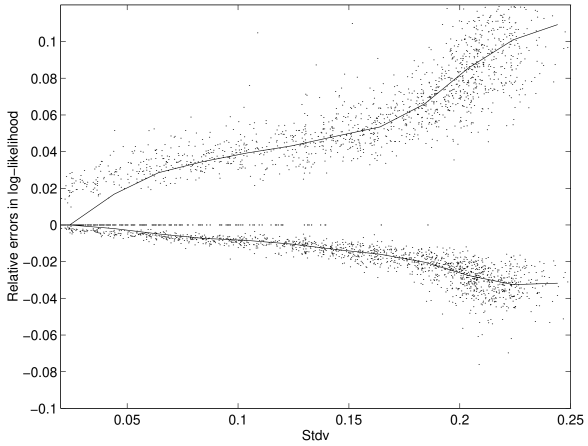

Figure 2: Accuracy of the bounds for noisy-OR networks. The solid lines are the median relative errors in log-likelihood as a function of σ_{std}. The upper and lower curves correspond to the upper and lower bounds respectively.

Toward extending the work presented in this paper we note that both the upper and lower bounds can be improved by considering a mixture paritioning (Jaakkola & Jordan, 1996) of the space of uninstantiated variables instead of using a completely factorized approximation. Furthermore, the restriction of the upper bounds for two-level networks can be overcome, for example, by interlacing them with sampling techniques, although other extensions may be possible as well. Following Saul & Jordan (1996) we may also merge the obtained bounds with exact methods whenever they are feasible.

Acknowledgments

The authors wish to thank L. K. Saul for useful comments.

References

P. Dayan, G. Hinton, R. Neal, and R. Zemel (1995). The Helmholtz machine. *Neural Computation* **7**: 889-904.

P. Dagum and M. Luby (1993). Approximate probabilistic reasoning in Bayesian belief networks is NP-hard. *Artificial Intelligence* **60**: 141-153.

G. Hinton, P. Dayan, B. Frey, and R. Neal (1995). The wake-sleep algorithm for unsupervised neural networks. *Science* **268**: 1158-1161.

T. Jaakkola, L. Saul, and M. Jordan (1996). Fast learning by bounding likelihoods in sigmoid-type belief networks. To appear in *Advances of Neural Information Processing Systems 8. MIT Press.*

T. Jaakkola and M. Jordan (1996). Mixture model approximations for belief networks. Manuscript in preparation.

F. V. Jensen, S. L. Lauritzen, and K. G. Olesen (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly* **4**: 269-282.

S. L. Lauritzen and D. J. Spiegelhalter (1988). Local computations with probabilities on graphical structures and their application to expert systems. *J. Roy. Statist. Soc. B* **50**:154-227.

P. McCullagh & J. A. Nelder (1983). *Generalized linear models.* London: Chapman and Hall.

R. Neal. Connectionist learning of belief networks (1992). *Artificial Intelligence* **56**: 71-113.

J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann: San Mateo.

L. K. Saul, T. Jaakkola, and M. I. Jordan (1996). Mean field theory for sigmoid belief networks. To appear in *JAIR.*

L. Saul and M. Jordan (1996). Exploiting tractable substructures in intractable networks. To appear in *Advances of Neural Information Processing Systems 8. MIT Press.*

M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz. H. P. Lehmann, G. F. Cooper (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Meth. Inform. Med.* **30**: 241-255.

A Sigmoid transformation

Here we derive and discuss the following transformation:

$$g(x) = \frac{1}{1+e^{-x}} = \min_{\xi \in [0,1]} e^{\xi x - H(\xi)}$$

Although a proof by hindsight would be shorter than a direct derivation we present the derivation for it is more informative. To this end, let us switch to log scale and consider

$$\begin{aligned}
-\log(1+e^{-x}) &= -\log \sum_{m\in\{0,1\}} e^{-mx} \\
&= -\log \sum_{m\in\{0,1\}} \xi^m (1-\xi)^{1-m} \frac{e^{-mx}}{\xi^m(1-\xi)^{1-m}} \\
&= -\log E\{\frac{e^{-mx}}{\xi^m(1-\xi)^{1-m}}\} \\
&\leq E\{-\log \frac{e^{-mx}}{\xi^m(1-\xi)^{1-m}}\} \\
&= \xi x + \xi \log \xi + (1-\xi)\log(1-\xi) \\
&= \xi x - H(\xi)
\end{aligned}$$

which follows from interpreting $\xi^m(1-\xi)^{1-m}$ as a probability mass for m and from an application of Jensen's inequality. By actually performing the minimization over ξ gives $\xi^* = g(-x)$ and leads to an equality instead of a bound. The geometry of the bound when ξ is kept fixed for all x is illustrated in figure 3. The value of x for which the chosen ξ is optimal is the point where the bound is exact.

We finally note that the above transformation can be understood as a type of Legendre transformation.

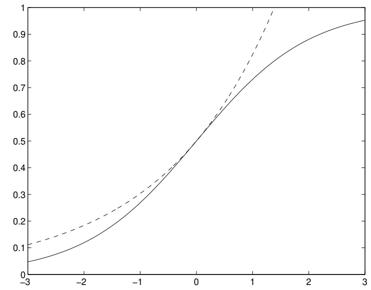

Figure 3: Geometry of the sigmoid transformation. The dashed curve plots $\exp\{\xi x - H(\xi)\}$ as a function of x for a fixed ξ (=0.5).

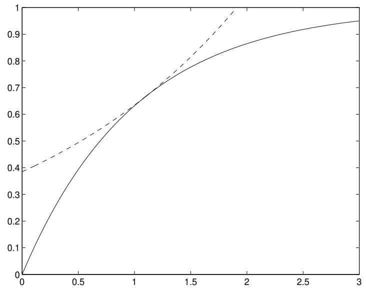

Figure 4: Geometry of the noisy-OR transformation. The dashed curve gives $\exp\{\xi x - F(\xi)\}$ as a function of x when ξ is fixed at 0.5.

B Noisy-OR transformation

Here we provide a derivation for the transformation

$$1 - e^{-x} = \min_{\xi \geq 1} e^{\xi x - F(\xi)} \tag{39}$$

presented in the text. Switching to log scale we find

$$\begin{aligned}
\log(1-e^{-x}) = -\log \frac{1}{1-e^{-x}} &= -\log \sum_{k=0}^{\infty} e^{-kx} \\
&= -\log \sum_{k=0}^{\infty} (1-q)q^k \frac{e^{-kx}}{(1-q)q^k} \\
&= -\log E\{\frac{e^{-kx}}{(1-q)q^k}\} \\
&\leq E\{-\log \frac{e^{-kx}}{(1-q)q^k}\} \\
&= \sum_{k=0}^{\infty} (1-q)q^k kx + \sum_{k=0}^{\infty} (1-q)q^k [\log(1-q) + k \log q] \\
&= \frac{q}{1-q} x + \log(1-q) + \frac{q}{1-q} \log q
\end{aligned}$$

where we have interpreted $(1-q)q^k$ as a probability distribution for k and used Jensen's inequality. Minimizing the above bound with respect to q gives $q^* = e^{-x}$ and the bound becomes exact. The original transformation follows by setting $\xi = q/(1-q)$. If the value of ξ is kept constant, the transformation yields a bound, the geometry of which is shown in figure 4. The point where the bound touches the $1 - e^{-x}$ curve defines x for which the constant ξ is optimal.

As in the sigmoid case the resulting transformation can be seen as a type of Legendre transformation.

C Noisy-OR expansion

The noisy-OR expansion

$$1 - e^{-x} = \prod_{k=0}^{\infty} g(2^k x) \tag{40}$$

follows simply from

$$\begin{aligned}
1 - e^{-x} &= \frac{(1+e^{-x})(1-e^{-x})}{1+e^{-x}} \\
&= g(x)(1-e^{-2x}) \\
&= g(x)\frac{(1+e^{-2x})(1-e^{-2x})}{1+e^{-2x}} \\
&= g(x)g(2x)(1-e^{-4x})
\end{aligned} \tag{41}$$

and induction. For $x > 0$ the accuracy of the expansion is governed by $1 - e^{-2^k x}$ which goes to one exponentially fast. Also since $g(2^k 0) = 1/2$, the expansion becomes $(\frac{1}{2})^N$ at $x = 0$, where N is the number of terms included. As this approaches $1 - e^{-0} = 0$ exponentially fast, we conclude that the rapid convergence is uniform. Figure 5 illustrates the accuracy of the expansion for small N.

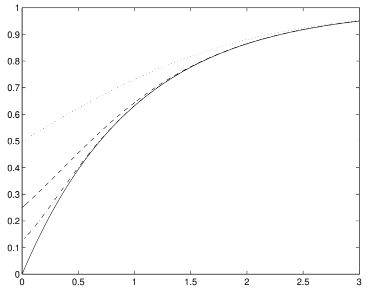

Figure 5: Accuracy of the noisy-OR expansion. Dotted line: $N = 1$, dashed line: $N = 2$, dotdashed: $N = 3$. N is the number of terms included in the expansion.

D Quadratic bound

For $X \in [0,1]$ we can bound $-\log(1+X)$ by a quadratic expression:

$$-\log(1+X) \geq a(X-x)^2 + b(X-x) + c \tag{42}$$

where $c = -\log(1+x)$, $b = -1/(1+x)$, and $a = -[(1-x)b + c + \log 2]/(1-x)^2$. The coefficents can be derived by requiring that the quadratic expression and it's derivative are exact at $X = x$, and by choosing the largest possible a such that the expression remains a bound. The resulting approximation is good for all $x \in [0,1]$ and can be optimized by setting $x = E\{X\}$.

Let us now use this quadratic bound in eq. (36) to better approximate the expectations. To simplify the ensuing formulas we use the notation

$$\begin{aligned} & E_Q\{\, e^{-2^k \sum_j \theta_{ij} S_j} \,\} = \\ & = \prod_j \left(\mu_j e^{-2^k \theta_{ij}} + 1 - \mu_j \right) = X_i^{(k)} \end{aligned} \quad (43)$$

With these we straightforwardly find

$$\begin{aligned} & \log P(\{S_i\}_{i\in L}|\theta) \\ & \geq \sum_{ik} \mu_i a_{ik} \left[X_i^{(k+1)} - 2X_i^{(k)} x_i^{(k)} + (x_i^{(k)})^2 \right] \\ & \quad + \sum_{ik} \mu_i \left[b_{ik}(X_i^{(k)} - x_i^{(k)}) + c_{ik} \right] \\ & \quad - \sum_i (1-\mu_i) \sum_j \theta_{ij}\mu_j + H_Q \end{aligned} \quad (44)$$

which is optimized with respect to $x_i^{(k)}$ simply by setting $x_i^{(k)} = X_i^{(k)}$. The simpler bound in eq. (37) corresponds to ignoring the quadratic correction, i.e., using $a_{ik} = 0$ above.