Austrian Research Institue for Artificial Intelligence
Österreichisches Forschungsinstitut für Artificial Intelligence
http://www.oefai.at/oefai/oefai.html

Schottengasse 3, Vienna
A-1010, Austria
(+43-1) 5336112-0

ÖFAI Neural Computation Group Technical Report

October 16, 2003

OEFAI–TR–2003–29

Variational Action Selection for Influence Diagrams

Brian Sallans
ÖFAI Neural Computation Group

Abstract

Influence diagrams provide a compact way to represent problems of decision making under uncertainty. As the number of variables in the problem increases, computing exact expectations and making optimal decisions becomes computationally intractable. A new method of action selection is presented, based on variational approximate inference. A policy is approximated where high-probability actions under the policy have high utility. Actions are then selected which have high probability under the approximating policy. The variational action selection method is shown to compare favorably to greedy and sampling-based action selection.

Variational Action Selection for Influence Diagrams

Brian Sallans
ÖFAI Neural Computation Group

1 Introduction

Making decisions under uncertainty involves inferring the state of the world based on the current evidence, and then selecting actions to maximize the expected benefit to the actor. An influence diagram provides a compact way to represent such a decision problem [III, Merkhofer, Howard, Matheson, and Rice 1976; Howard and Matheson 1984]. It includes a graphical structure, definitions of probabilities over the state variables in the system, the actions available, and utilities which quantify benefits to the actor.

The graphical portion of an influence diagram consists of nodes and directed edges between nodes. The directed acyclic graph shows what state variables and actions are under consideration, and their relationships to one another and to the utility function. An influence diagram can include three types of nodes: chance nodes, decision nodes, and utility nodes. The nodes represent (observed and unobserved) random variables, actions, and utilities respectively. The lack of an edge between two chance nodes indicates independence. The actor decides the settings of the decision nodes, which can influence utilities or the distributions over chance nodes. The parents of a decision node indicate the information available at the time that the decision must be made. The parents of a utility node show which variables directly impact utility.

Evaluating an influence diagram involves finding a policy which maximizes expected utility given the evidence. The policy is a mapping from settings of the evidence nodes to actions or distributions over actions. Computing expectations and selecting good actions is intractable for general, densely connected influence diagrams with a large number of variables. In this article we present a method for selecting good actions when computation of the optimal policy is intractable. We restrict ourselves to influence diagrams where all decisions are made simultaneously, and do not have an impact on distributions over unobserved chance nodes. We also assume that the total utility is found by adding individual utility node values (see Figure 1).

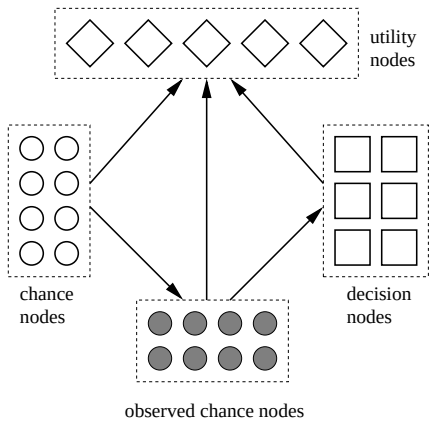

Figure 1: A generic influence diagram of the form that we consider. Shaded circles indicate observed chance nodes, and squares indicate decision nodes. The unobserved chance nodes are indicated by open circles. The utility nodes are diamonds. Total utility is found by adding the values of all utility nodes.

There are a number of algorithms for the exact evaluation of influence diagrams [Howard and Matheson 1984]-[Zhang 1998]. In the interest of brevity, we will not detail them there. There are two general categories of exact algorithms: ones that operate directly on the influence diagram [Shachter 1986;

Shachter and Kenley 1989], and ones that first transform it into an intermediate representation [Howard and Matheson 1984; Cooper 1988; Shachter and Peot 1992; Zhang 1998]. The former results in a dynamic programming approach, where the optimal policy is found by working backwards from utility nodes, eliminating decision nodes and integrating out chance nodes. The latter involves performing inference in a sequence of Bayesian networks [Cooper 1988; Shachter and Peot 1992; Zhang 1998], or operating on a decision tree [Howard and Matheson 1984].

Approximate inference techniques such as sampling methods can also be used to evaluate the expectations required to do planning [Charnes and Shenoy 1999; Ortiz and Kaelbling 2000a; Ortiz and Kaelbling 2000b] or to select the actions themselves [Bielza, Müller, and Insua 1999]. In this case, a full policy is not necessarily found. Instead of finding a full mapping from settings of the evidence nodes to actions, an optimal action can be found for a particular setting of the evidence nodes. This is appropriate when there is no compact representation of the full policy, or when we are only interested in making decisions for particular instantiations of the evidence. This is the approach that we will take.

Variational methods are an alternative to sampling for approximating intractable distributions. Variational inference has been used for approximate inference in a wide variety of graphical models [Saul, Jaakkola, and Jordan 1996]-[Ghahramani and Beal 2000]. Variational methods are fast, deterministic, and can take advantage of structure in the graph to greatly speed up and improve inference. We present a method for using variational inference algorithms for approximate inference and planning in influence diagrams that are too large to be evaluated exactly. We show experimental results comparing the variational method to alternative methods including greedy and sampling-based action selection.

2 Variational Action Selection

As the number of decision nodes increases, finding the optimal action for a particular instantiation of evidence becomes intractable. Instead of finding the optimal action directly, we will estimate a particular stochastic policy. The policy will be defined so that an action with the highest probability under the policy is optimal. After estimating the stochastic policy, we can then select an action by finding the mode of the approximating policy.

Let the observed chance node settings be denoted by $\mathbf{o}$, the unobserved chance node settings by $\mathbf{h}$, and the decision node settings by $\mathbf{a}$. Bold face denotes a vector, and subscripts will denote elements of a vector. Consider a utility function $U(\mathbf{o}, \mathbf{h}, \mathbf{a}) > 0$, which is a positive function of the settings of the nodes. Given a policy $\pi(\mathbf{a}, \mathbf{o}) = P(\mathbf{a}|\mathbf{o})$ the expected utility is given by:

$$U^{\pi}(\mathbf{o}) = \sum_{\mathbf{h},\mathbf{a}} \pi(\mathbf{a}, \mathbf{o}) P(\mathbf{h}|\mathbf{a}, \mathbf{o}) U(\mathbf{o}, \mathbf{h}, \mathbf{a}) \tag{1}$$

We are interested in a particular policy defined as:

$$\pi(\mathbf{a}, \mathbf{o}) = \frac{\sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{a}, \mathbf{o}) U(\mathbf{o}, \mathbf{h}, \mathbf{a})}{\sum_{\mathbf{h},\mathbf{a}'} P(\mathbf{h}|\mathbf{a}', \mathbf{o}) U(\mathbf{o}, \mathbf{h}, \mathbf{a}')} \tag{2}$$

The policy has been defined so that an action which maximizes probability under the policy is an optimal action.

With a large number of hidden chance and decision nodes, this policy would be at least as difficult to find as the optimal action. Instead, we will select an approximation to $\pi(\mathbf{a}, \mathbf{o})$ from a restricted class of distributions Ω. The class Ω will be chosen so that any distribution in Ω can be represented compactly, its mode can be found efficiently, and expectations of the log-utility function can be evaluated quickly.

Given an arbitrary element $\varpi \in \Omega$, we can approximate the log-utility. For the remainder of this section, we assume that the unobserved chance nodes are absorbed into the utility nodes in a preprocessing step, yielding a new utility function $U(\mathbf{o}, \mathbf{a})$. Starting with the (negative) variational free energy (with the log utility corresponding to the negative energy function), we approximate the log utility as

follows:

$$\begin{aligned}
\mathcal{F} &= \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log U(\mathbf{o}, \mathbf{a}) - \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \varpi(\mathbf{a}) && (3)\\
&= \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \frac{U(\mathbf{o}, \mathbf{a})(\sum_{\mathbf{a}} U(\mathbf{o}, \mathbf{a}))}{\sum_{\mathbf{a}} U(\mathbf{o}, \mathbf{a})} - \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \varpi(\mathbf{a}) && (4)\\
&= \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \frac{\pi(\mathbf{o}, \mathbf{a}) U^{\pi}(\mathbf{o})}{\rho} - \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \varpi(\mathbf{a}) && (5)\\
&= \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \frac{U^{\pi}(\mathbf{o})}{\rho} - \sum_{\mathbf{a}} \varpi(\mathbf{a}) \log \frac{\varpi(\mathbf{a})}{\pi(\mathbf{o}, \mathbf{a})} && (6)\\
&= \log U^{\pi}(\mathbf{o}) - \log \rho - \mathrm{D}(\varpi \| \pi) && (7)
\end{aligned}$$

where $\mathrm{D}(\cdot\|\cdot)$ is the Kullback-Leibler divergence between two distributions, and ρ is defined as: $\rho = U^{\pi}(\mathbf{o}) / \sum_{\mathbf{a}} U(\mathbf{o}, \mathbf{a})$.

This is essentially the same reasoning that leads to variational lower bounds on the log-likelihood [Neal and Hinton 1998; Jordan, Ghahramani, Jaakkola, and Saul 1998], except that the joint probability has been replaced by utility. Note that in this case a lower bound on the log utility is not maintained, due to the presence of ρ. However, since our motivation is different from the case of density estimation, maintaining the lower bound is not important. In this case, we are not interested in evaluating expected log-utilities with respect to π. The goal is to find an approximation to the policy π, whose mode will correspond to a good action. Maximizing Eq.(3) with respect to the parameters of the approximating policy ϖ minimizes the Kullback-Leibler divergence between the approximating and exact policies (Eq.(7)). We can then select an action by finding the mode $\tilde{\mathbf{a}} = \arg\max_{\mathbf{a}} \varpi(\mathbf{a})$.

3 Action Selection with Unobserved Chance Nodes

In the above we assume that it is tractable to evaluate the expected utility with respect to the unobserved chance nodes. This will not be the case in general when the influence diagram has a large number of hidden chance nodes.

Given an instantiation of the evidence, we can use an approximate inference algorithm to evaluate expected utilities. Sampling methods have been used for this purpose [Bielza, Müller, and Insua 1999; Ortiz and Kaelbling 2000a; Ortiz and Kaelbling 2000b].

We will use a variational inference method instead. The variational method can be used to lower-bound the marginal log-likelihood. An approximating distribution Q is selected so that evaluating this lower bound is tractable. The free energy can then be maximized with respect to Q, yielding an approximate posterior.

In our case we need to evaluate two expectations: the expected complete-data log-likelihood and the expected utility. We will therefore select an approximating distribution such that both of these quantities can be efficiently evaluated. Given the expectations, we can then select actions as described above.

4 Experimental Results

The variational action selection method was tested on several artificial influence diagrams. The first and second sets of tests were for the case of exact inference over the chance nodes. The third set of tests combined approximate action selection with approximate inference.

For simplicity, we restricted ourselves to N binary decision nodes and N binary chance nodes as parents of a set of additive utility nodes. We also used a single binary evidence node with the N unobserved chance nodes as parents (see Figure 2).

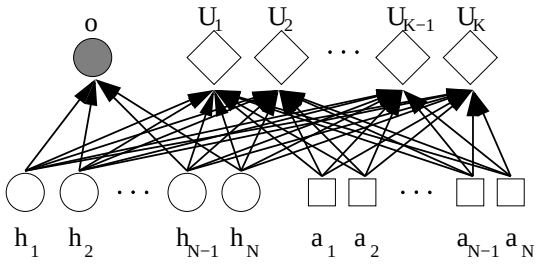

Figure 2: The influence diagram used for testing. There is a single (shaded) evidence node o, K utility nodes, N unobserved binary chance nodes and N binary decision nodes.

The utilities were randomly initialized for each setting of the parent nodes, by drawing them from a uniform distribution on the interval $(0, 100]$. Similarly, the likelihood of the evidence was randomly initialized for each setting of the parent nodes, drawn from a uniform distribution on $(0, 1]$.

4.1 Exact Inference

Here we assume that exact inference can be performed over the unobserved chance nodes. The intent is to test variational action selection, without the results being obscured by approximate inference.

We tested three algorithms: greedy, sampling-based [Bielza, Müller, and Insua 1999] and variational action selection. Greedy action selection was implemented by randomly initializing the decision nodes, and then iterating repeatedly, setting each decision node to maximize utility given the others. The iteration was terminated when a local maximum was reached. The sampling algorithm used Gibbs sampling to select an action conditioned on all other actions. Given the action settings $\{a_i\}_{i=1,i\neq k}^N$, action a_k was set to one with probability:

$$P(a_k = 1) \leftarrow \sigma\left(\log U(\mathbf{o}, \{a_i\}_{i=1,i\neq k}^N, a_k = 1) - \log U(\mathbf{o}, \{a_i\}_{i=1,i\neq k}^N, a_k = 0)\right) \tag{8}$$

where $\sigma(x) = 1/(1 + \exp\{-x\})$. After sampling actions for a fixed number of iterations, the approximate mode was taken to be the best action in the sample set.

The variational method used a fully-factored approximating distribution:

$$\varpi(\mathbf{a}) = \prod_{i=1}^{N} \varpi_i^{a_i}(1 - \varpi_i)^{(1-a_i)} \tag{9}$$

An update equation for each parameter ϖ_k was found by differentiating the negative free energy (Eq.(3)) with respect to ϖ_k and setting to zero. During optimization, the parameters ϖ_k were restricted to be near valid action values. The parameters could not be restricted to take on the values of zero or one, because the entropy term in Eq.(3) requires that probabilities be non-zero. Instead they were restricted to take on values of 0.99 or 0.01. The update equation for ϖ_k was:

$$\varpi_k \longleftarrow \begin{cases} 0.99 & \text{if } S_k \geq 0.5; \\ 0.01 & \text{otherwise.} \end{cases} \tag{10}$$

where S_k is defined as:

$$S_k = \sigma\left(\sum_{\{\mathbf{a}:a_k=1\}} \frac{\varpi(\mathbf{a})}{\varpi_k} \log U(\mathbf{o}, \mathbf{a}) \; - \sum_{\{\mathbf{a}:a_k=0\}} \frac{\varpi(\mathbf{a})}{1 - \varpi_k} \log U(\mathbf{o}, \mathbf{a})\right) \tag{11}$$

As with the greedy method, the update equations were iterated until convergence. After convergence, an action was selected by thresholding the parameter values:

$$a_k \longleftarrow \begin{cases} 1 & \text{if } \varpi_k \geq 0.5; \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The first set of tests were conducted on 1000 randomly generated influence diagrams, each of which had a single utility node with all of the action nodes as its parents. The mean results and 95% confidence intervals on the mean are shown in Figure 3(a) for different numbers of action nodes. Relative error was computed as $\mathrm{RE} = (U_{\mathrm{opt}} - U_{\mathrm{approx}})/U_{\mathrm{opt}}$, where U_{opt} and U_{approx} are the utilities achieved with the optimal and approximated actions respectively. On these artificial influence diagrams a uniform random policy has a relative error of approximately 0.5.

The second set of tests were conducted on 1000 randomly generated influence diagrams, each of which had 100 binary action nodes and 20 (additive) utility nodes. Each utility nodes had P randomly selected action nodes as parents. In this test, there were far too many action nodes to compute optimal actions exactly. Instead we simply compare the utility achieved by each of the three approximation techniques for different numbers of parents. The mean results and 95% confidence interval on the mean are shown in figure 3(b).

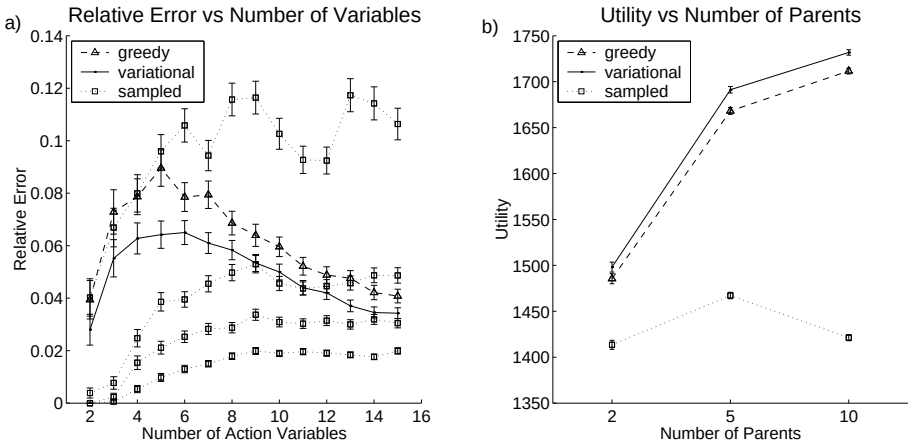

Figure 3: (a) Relative loss of utility vs. number of binary decision nodes for different action selection techniques. The dotted curves show the sampling method with (from top to bottom) 5, 10, 15 and 25 samples. b) Utility vs. number of utility node parents, for 20 utility nodes and 100 binary action nodes. In all tests, exact expectations were known. The solid curves show the variational method. The dashed curves show the greedy method. The dotted curves show the sampling method.

In the single-utility-node case, both the variational method and the greedy method converged after less than three iterations on average, with none taking longer than seven iterations. The variational method achieved significantly better average utility than the greedy method on all network sizes greater than three.[1] The utility achieved by the sampling method depended on the number of samples taken. Performance matched the other methods after only five samples for small networks. On larger networks similar performance was reached with 10 to 15 samples. Results are also shown for 25 samples, which outperformed the other two methods for all networks. Collecting a single sample action required approximately the same amount of computation as one iteration of the other algorithms.

In the multi-utility-node case, the variational method significantly outperformed the other two methods. The sampling method fared worse in these tests, even though a sample size of 50 was used in all cases. The greedy and variational methods converged after 3.6 iterations on average, with no run taking more than 6 iterations.

4.2 Approximate Inference

In these experiments we combined a variational approximate inference technique with variational action selection. Each influence diagram had N binary unobserved chance nodes, N binary decision nodes, and a single utility node. The likelihood function was randomly initialized to produce a probability for

[1] All statistical significance testing was done using a Wilcoxon signed rank test at significance $p = 0.05$.

each setting of the parent chance nodes. Similarly, a utility function was randomly generated to produce values in the range $(0, 100]$ (chosen from the uniform distribution) for each setting of the chance and decision nodes.

The variational inference technique used a fully factored approximating distribution:

$$Q(\mathbf{h}) = \prod_i \mu_i^{h_i} (1 - \mu_i)^{(1-h_i)} \tag{13}$$

An update equation for each variational parameter was produced by differentiating the variational free energy, setting the derivative to zero and solving for the parameter. The update equation for parameter μ_k is given by:

$$\mu_k \longleftarrow \sigma \left(\sum_{\{\mathbf{h}: h_k = 1\}} \frac{\mu(\mathbf{h})}{\mu_k} \log P(\mathbf{o}, \mathbf{h}) \; - \sum_{\{\mathbf{h}: h_k = 0\}} \frac{\mu(\mathbf{h})}{1 - \mu_k} \log P(\mathbf{o}, \mathbf{h}) \right) \tag{14}$$

Each method was tested on 1000 randomly generated influence diagrams. The update equations were iterated until convergence. This took only a few iterations in each case. The results are shown in Figure 4.

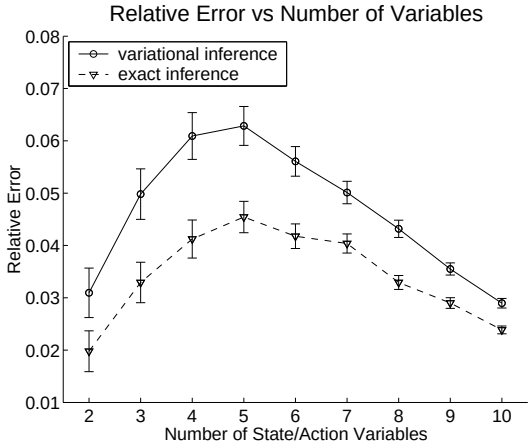

Figure 4: Relative loss of utility for variational action selection with (upper curve) and without (lower curve) variational approximate inference. The method with approximate inference is uniformly worse, but the error decreases for larger numbers of decision and chance nodes.

Action selection with approximate inference suffers as compared to exact inference, with performance dropping in the worst case by 36.1% (for $N = 2$). In the best case, the decrease in performance is 17.6% (for $N = 10$). In general, the performance of the variational approximation improves as the number of nodes increase.

5 Discussion

We have described a general method of decision making in a subset of large influence diagrams. Instead of finding an optimal action directly, we approximate a policy which gives maximum probability to optimal actions. We then select an action by finding the mode of this approximating policy. Although we have focused on discrete random variables, there is no reason why the same technique could not also be applied to influence diagrams with continuous nodes. The difficulty, as with any variational method, lies in finding a good approximating distribution.

The experimental results indicate that variational action selection is a competitive technique, especially when computational resources are limited. The sampling method was found to perform better for larger sample sizes on some networks. However, larger sample sizes required correspondingly more

computational effort to collect. This was to be expected, based on results of using the two methods for inference in graphical models: Although sampling methods are guaranteed to get the right answer in the limit, they can be computationally demanding. In the case of Markov chain Monte Carlo methods assessing convergence can also be problematic. Variational methods are comparatively fast, deterministic, and convergence of parameter updates can be assessed. However, there are no *a priori* performance guarantees. The performance of a particular variational approximation must be tested empirically on the problem of interest. The results on our set of artificial influence diagrams are encouraging. Variational action selection warrants further investigation as an alternative method of action selection in large influence diagrams.

Acknowledgments

The author would like to thank Zoubin Ghahramani and Peter Dayan for useful discussions. This work was funded by the Austrian Science Fund (FWF) under grant SFB#010: "Adaptive Information Systems and Modeling in Economics and Management Science". The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Education, Science and Culture and by the Austrian Federal Ministry for Transport, Innovation and Technology.

References

Bielza, C., P. Müller, and D. R. Insua (1999). Decision analysis by augmented probability simulation. *Management Science 45*(7), 995–1007.

Charnes, J. M. and P. P. Shenoy (1999). A forward Monte Carlo method for solving influence diagrams using local computation. School of Business Working Paper No. 273, School of Business, University of Kansas.

Cooper, G. F. (1988). A method for using belief networks as influence diagrams. In *Proc. of the Fourth Conference on Uncertainty in Artificial Intelligence*, pp. 55–63.

Ghahramani, Z. and M. Beal (2000). Variational inference for Bayesian mixtures of factor analysers. In S. A. Solla, T. K. Leen, and K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems*, Volume 12, pp. 449–455. The MIT Press, Cambridge.

Ghahramani, Z. and G. E. Hinton (1998). Variational learning for switching state-space models. *Neural Computation 4*(12), 963–996.

Ghahramani, Z. and M. I. Jordan (1997). Factorial hidden Markov models. *Machine Learning 29*, 245–273.

Howard, R. A. and J. E. Matheson (1984). *The Principles and Applications of Decision Analysis*, Volume II, Chapter Influence Diagrams, pp. 690–718. Strategic Decision Group, Mento Park, CA.

III, A. M., M. Merkhofer, R. Howard, J. Matheson, and T. Rice (1976). Development of decision aids for decision analysis. Technical report, Stanford Research Institute, Menlo Park, CA.

Jaakkola, T. S. (1997). *Variational Methods for Inference and Estimation in Graphical Models*. Cambridge, MA: Department of Brain and Cognitive Sciences, MIT. Ph.D. thesis.

Jordan, M., Z. Ghahramani, T. Jaakkola, and L. Saul (1998). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in Graphical Models*, pp. 105–161. Kluwer Academic Publishers.

Neal, R. M. and G. E. Hinton (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in Graphical Models*, pp. 355–368. Kluwer Academic Publishers.

Ortiz, L. and L. P. Kaelbling (2000a). Sampling methods for action selection in influence diagrams. In *Proc. Seventeenth National Conference on Artificial Intelligence*.

Ortiz, L. E. and L. P. Kaelbling (2000b). Adaptive importance sampling for estimation in structured domains. In *Proc. of Uncertainty in AI (UAI-00), 2000.*

Saul, L. K., T. S. Jaakkola, and M. I. Jordan (1996). Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research 4*, 61–76.

Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research 34*, 871–882.

Shachter, R. D. and C. R. Kenley (1989). Gaussian influence diagrams. *Management Science 35*, 527–550.

Shachter, R. D. and M. A. Peot (1992). Decision making using probabilistic inference methods. In *Proc. of the Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 276–283.

Zhang, N. L. (1998). Probabilistic inference in influence diagrams. In *Proc. UAI'98*, pp. 514–522.