Bayesian Networks

8. Approximate Inference / Adaptive Importance Sampling and Loopy Propagation

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

Problems of Likelihood Weighting Sampling

Likelihood weighting sampling still can reject cases, if the cdfs of the evidence variables have zeros and thus can generate a case weight 0.

Example: consider the studfarm example with evidence $J = \text{AA}$ again. Whenever H or I are pure (aa), J cannot be sick. In these cases the case weight is zero, e.g.,

$$w(x) := p_J(J = \text{AA}|H = \text{aa}, I = \ldots) = 0$$

and the sample is dropped.

H	aa		aA	
I	aa	aA	aa	aA
J= aa	1	.5	.5	.25
aA	0	.5	.5	.5
AA	0	0	0	.25

Figure 1: Studfarm example: $p(J|H, I)$ if H and I cannot be sick.

As the marginal of H, I w/o evidence is

I	aa	aA
H = aa	0.98265	0.00823
aA	0.00742	0.00170

the probability for acceptance is only

$$p(H = \text{aA}, I = \text{aA}) = 0.00170$$

i.e., only 1 from 588 samples is accepted.

Some rejections may be unavoidable

If CPDs have zeros, forward sampling always may lead to some rejected cases.

Example 1. If we observe evidence

$$C = 1,$$

then

$$p(A = 0|C = 1) > 0$$

and

$$p(B = 0|C = 1) > 0,$$

thus forward sampling

(i) will have to sample $A = 0$ as well as $B = 0$,

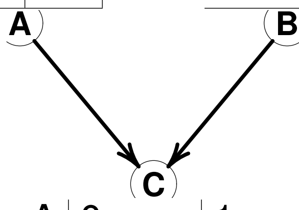
(ii) will sample A and B independently, and thus

(iii) will occasionally sample $A = 0$ and $B = 0$,

which will be rejected as it is not compatible with the observed evidence.

A = 0	0.1
1	0.9

B = 0	0.7
1	0.3



A	0		1	
B	0	1	0	1
C = 0	1	0.7	0.3	0.6
1	0	0.3	0.7	0.4

Figure 2: Bayesian network with a zero in a conditional potential.

Optimal sampling distribution

Theorem 1 (Rubinstein 1981). *The optimal sampling distribution is* $q = p$.

i.e., in our case:

$$q = p_E = \prod_{v \in V} (p_v)_E$$

Idea of Self Importance Sampling:

(i) compute $(p_v)_E$ for all vertices $v \in V$,

(ii) sample from $q := p_E$ by replacing the vertex potentials p_v by $(p_v)_E$.

Forward sampling automatically samples from $(p_v)_E$ for all vertices v w/o. evidence descendant (as then all evidence vertices have been enumerated before v and we effectively sample conditional on all vertices sampled before).

$\Rightarrow$ $(p_v)_E$ has to be estimated only for ancestors v of evidential vertices.

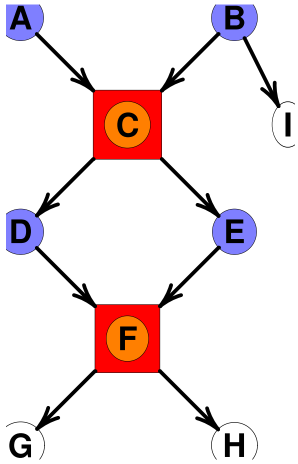


Figure 3: CPDs of blue vertices have to be estimated.

Self Importance Sampling [SP90]:

a) Update sampling distribution $q_v := \widehat{(p_v)_E}$ in step k:

$$\widehat{(p_v)_E}^{(k+1)} := (1 - \lambda) \cdot p_v + \lambda \cdot \widehat{(p_v)_E}^{(\text{all})}$$

with **learning rate**

$$\lambda(k) := \frac{k}{k+1}$$

where $\widehat{(p_v)_E}^{(\text{all})}$ is estimated based on all samples seen so far.

b) Estimate target potentials based on all samples generated.

Adaptive Importance Sampling [CD00]:

a) Update sampling distribution $q_v := \widehat{(p_v)_E}$ in step k:

$$\widehat{(p_v)_E}^{(0)} := p_v$$

$$\widehat{(p_v)_E}^{(k+1)} := (1 - \lambda) \cdot \widehat{(p_v)_E}^{(k)} + \lambda \cdot \widehat{(p_v)_E}^{(\text{new})}$$

with **learning rate**

$$\lambda(k) := \lambda_0 \cdot \left(\frac{\lambda_{\text{max}}}{\lambda_0}\right)^{k/k_{\text{max}}}$$

(with $\lambda_0 := 0.4$ and $\lambda_{\text{max}} := 0.14$) where $\widehat{(p_v)_E}^{(\text{new})}$ is estimated based on a fresh sample.

b) Estimate target potentials based on samples weighted by a factor dependend on step k (e.g., only on samples drawn in the last step).

Self Importance Sampling (SIS)

infer-sis$(B := (G, (p_v)_{v \in V_G}), W : \textit{target domain}, E : \textit{evidence},$
$\quad n : \textit{sample size}, k_{\max} : \textit{no of adaptions}, \lambda : \textit{learning rate}) :$
$(D, w) := 0$
$A := \textit{anc}(\text{dom}(E))$
$q_v := p_v, \quad \forall v \in V_G$
for $k := 1, \ldots, k_{\max}$ **do**
$\quad (D, w) := (D, w) \cup (\textit{sample-lw-tweaked}(B, (q_v)_{v \in V_G}, E) \mid i = 1, \ldots, \lfloor \frac{n}{k_{\max}} \rfloor)$
$\quad (\widehat{(p_v)_E}^{(\text{all})})_{v \in A} := \textit{estimate}(D, w, \{\text{dom}(p_v) \mid v \in A\})$
$\quad q_v := (1 - \lambda(k)) \cdot p_v + \lambda(k) \cdot \widehat{(p_v)_E}^{(\text{all})}, \quad \forall v \in A$
od
return $\textit{estimate}(D, w, W)$

sample-lw-tweaked$(B := (G, (p_v)_{v \in V_G}), (q_v)_{v \in V_G \setminus \text{dom}(E)} : \textit{sampling distribution}, E : \textit{evidence}) :$
$\sigma := \textit{topological-ordering}(G \setminus \text{dom}(E))$
$x := 0_{V_G}$
$x|_{\text{dom}(E)} := \text{val}(E)$
for $i = 1, \ldots, |\sigma|$ **do**
$\quad v := \sigma(i)$
$\quad q := q_v|_{x|_{\text{pa}(v)}}$
$\quad \text{draw } x_v \sim q$
od
$w(x) := \prod_{v \in \text{dom}(E)} p_v(x_v \mid x|_{\text{pa}(v)}) \cdot \prod_{\substack{v \in V_G \setminus \text{dom}(E) \\ q_v \neq p_v}} \frac{p_v(x_v \mid x|_{\text{pa}(v)})}{q_v(x_v \mid x|_{\text{pa}(v)})}$
return $(x, w(x))$

Figure 4: Algorithm for approximate inference by Self Importance Sampling.

Adaptive Importance Sampling (AIS)

infer-ais$(B := (G, (p_v)_{v \in V_G}), W : \textit{target domain}, E : \textit{evidence},$
$\quad n : \textit{sample size}, k_{\max} : \textit{no of adaptions}, \lambda : \textit{learning rate}, \alpha : \textit{target weights}) :$
$(D, w) := 0$
$A := \textit{anc}(\text{dom}(E))$
$q_v := p_v, \quad \forall v \in V_G$
for $k := 0, \ldots, k_{\max}$ **do**
$\quad (D', w') := (\textit{sample-lw-tweaked}(B, (q_v)_{v \in V_G}, E) \mid i = 1, \ldots, \lfloor \frac{n}{k_{\max}+1} \rfloor)$
$\quad (D, w) := (D, w) \cup (D', w' \cdot \alpha(k))$
$\quad (\widehat{(p_v)_E}^{(\text{new})})_{v \in A} := \textit{estimate}(D', w', \{\text{dom}(p_v) \mid v \in A\})$
$\quad q_v := (1 - \lambda(k)) \cdot q_v + \lambda(k) \cdot \widehat{(p_v)_E}^{(\text{new})}, \quad \forall v \in A$
od
return $\textit{estimate}(D, w, W)$

Figure 5: Algorithm for approximate inference by Adaptive Importance Sampling.

[CD00] use $k_{\max} := 10$ and the targets weights

$$\alpha(k) := \begin{cases} 0, & \text{if } k < k_{\max} \\ 1, & \text{otherwise} \end{cases}$$

effectivly separating the estimation process for the sampling distribution and for the target potentials.

Measuring accuracy of estimates

To measure accuracy of estimated target potentials $\hat{p}_d$ $(d \in D)$ for a set of target domains D:

(i) for each target domain $d \in D$ the exact potential p_d is computed (e.g., by clustering),

(ii) the **mean squared error on parameters** is used as quality measure:

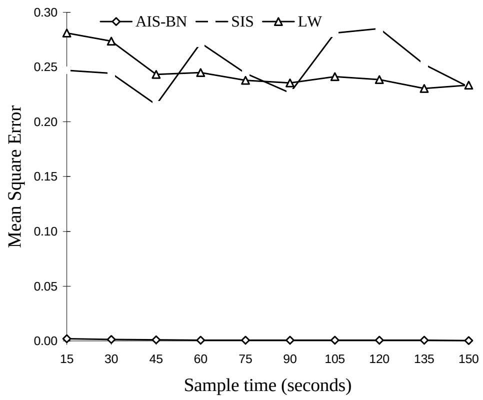


Figure 6: Experimental evaluation of LW, SIS, and AIS on CPCS network [CD00, p. 174].

$$\text{MSE}((\hat{p}_d)_{d\in D}) := \sqrt{\frac{1}{\sum_{d\in D} |\prod \text{dom}(d)|} \sum_{d\in D} \sum_{x \in \prod \text{dom}(d)} (\hat{p}_d(x) - p_d(x))^2}$$

As target domains usually all single variable domains are used.

[CD00] use as evidence the joint instantiation of 20 random leaf vertices.

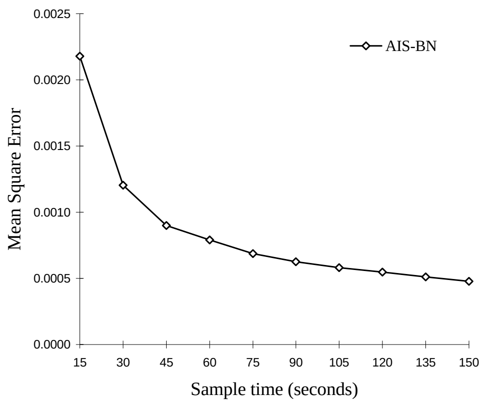


Figure 7: Convergence of AIS estimates: overall MSE [CD00, p. 175].

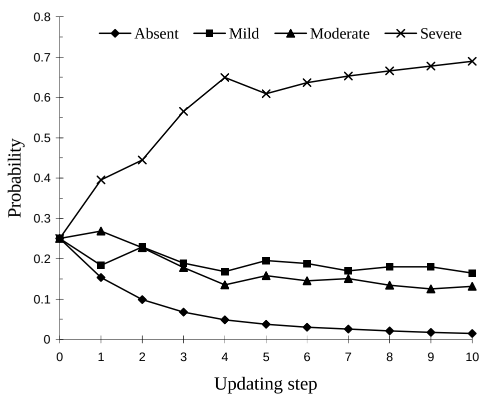


Figure 8: Convergence of AIS estimates for a single target potential [CD00, p. 176].

Heuristics for the improvement of importance sampling (1/2)

Two simple heuristics can dramatically improve the efficiency of the estimator [CD00]:

If the marginal probability of an evidential variable is low, i.e.,

$$p(X = e) < \frac{1}{2 \cdot |\operatorname{dom}(X)|}$$

then the vertex potentials of all its parent vertices are reset to a uniform distribution.

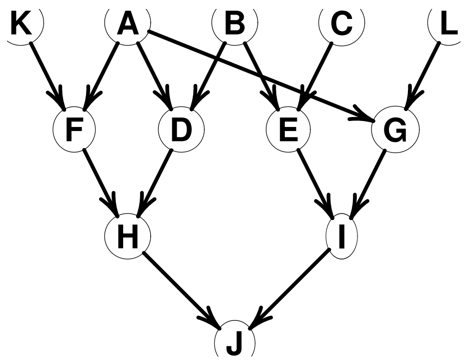


Figure 9: Studfarm bayesian network.

In the studfarm example

$$p(J = aa) = 0.00043 < \frac{1}{6}$$

thus $p(H|F,D)$ and $p(I|E,G)$ are reset to

father Y	aa		aA	
mother Z	aa	aA	aa	aA
aa	.5	.5	.5	.5
aA	.5	.5	.5	.5

Heuristics for the improvement of importance sampling (2/2)

Small coefficients of sampling potentials are replaced by a minimal threshold θ:

if $p_v(x|y) < \theta$ (for a $(x,y) \in \prod \operatorname{dom}(p_v)$), then

$$p_v(x|y)' := \theta$$
$$p_v(x'|y)' := p_v(x'|y) - (\theta - p_v(x|y)),$$

for x' with max. $p_v(x'|y)$

[CD00] use $\theta = 0.04$.

In the studfarm example, the probabilities of the root vertices will be adjusted:

A = aa	0.99	becomes	A = aa	0.96
aA	0.01		aA	0.04

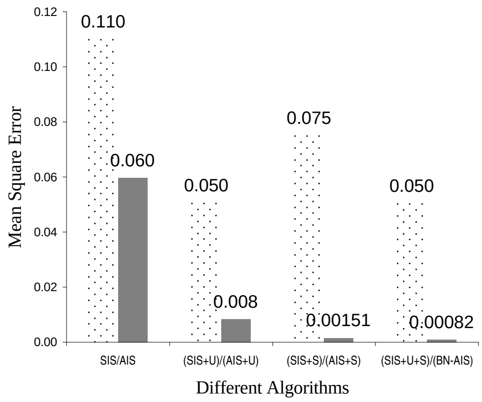


Figure 10: MSE of SIS and AIS with different initializations of the sampling distribution (stock p_v, with uniform parents (U), with small coefficients replaced (S), and with both) [CD00, p. 180].

1. Why exact inference may not be good enough
2. Acceptance-Rejection Sampling
3. Importance Sampling
4. Self and Adaptive Importance Sampling
5. **Stochastic / Loopy Propagation**

Cluster graphs

Definition 1. Let V be a set (of variables).
An undirected graph $G := (\mathcal{V}, E)$ on $\mathcal{V} \subseteq \mathcal{P}(V)$ is called an **cluster graph on** V, if

(i) the induced subgraph on all vertices containing a given variable v, i.e.,

$$\{W \in \mathcal{V} \mid v \in W\}$$

is connected for all variables $v \in V$.

and

(ii) all separators are non-empty

$$U \cap W \neq \emptyset, \quad \text{for all } U, W \in \mathcal{V}$$

Any cluster tree obviously is a cluster graph.

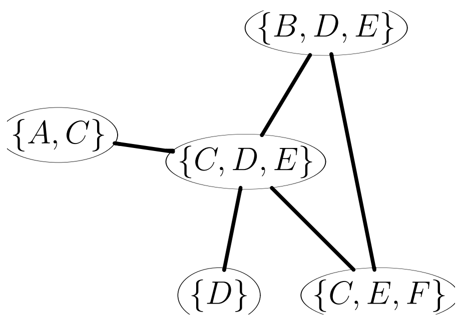


Figure 11: A cluster graph on $V := \{A, B, C, D, E, F\}$ that is not a cluster tree.

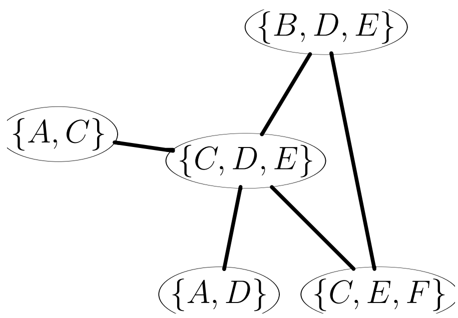


Figure 12: Not a cluster graph.

The family cluster graph

Let G be a directed graph. For $v \in V$

$$\mathrm{fam}(v) := \{v\} \cup \mathrm{pa}(v)$$

is called the **familiy of** v.

Let $(G = (V, E), (p_v)_{v \in V})$ be any Bayesian network (not necessarily a polytree). Let

$$\mathcal{V} := \{\mathrm{fam}(v) \,|\, v \in V\}$$

and

$$F := \{\{\mathrm{fam}(v), \mathrm{fam}(w)\} \,|\, v \in V, w \in \mathrm{pa}(v)\}$$

Then $H := (\mathcal{V}, F)$ is a cluster graph for $Q := \{p_v \,|\, v \in V\}$ called **family cluster graph**.

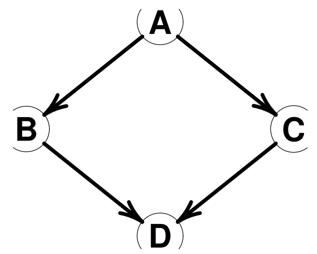


Figure 13: Bayesian network (that is not a polytree).

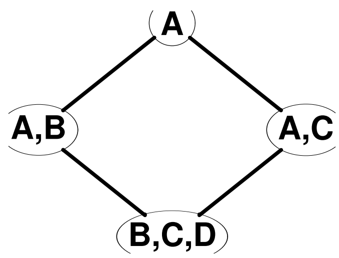


Figure 14: Family cluster graph of Bayesian network above.

Problem of loopy cluster graphs: there is no leaf to start computations with, but all link potentials depend on other linkpotentials.

Idea of loopy propagation:

(i) initialize link potentials to arbitrary values (uniform distribution; random distribution).

(ii) compute link potentials sucessively in arbitrary order.

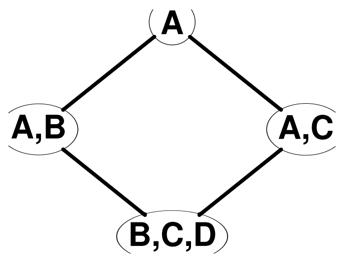


Figure 14: Family cluster graph of a Bayesian network.

This seems to be sensible in so far, as the true link potentials

$$q_{U,T} := p_U \prod_{\substack{W \in \mathrm{fan}(U) \\ W \neq T}} q_{W,U}$$

"often" form a fixpoint of the propagation operation, i.e., once all link potentials have their true values, any propagation step will reproduce the true value.

There are several arrangements of the computations possible:

Parallel loopy propagation [MWJ99]:
Compute

$$q_{U,T}^{(k+1)} := p_U \prod_{\substack{W \in \mathrm{fan}(U) \\ W \neq T}} q_{W,U}^{(k)}$$

in parallel for all U, T.

Sequential loopy propagation:
Fix an ordering of the links (U, T) and compute

$$q_{U,T} := p_U \prod_{\substack{W \in \mathrm{fan}(U) \\ W \neq T}} q_{W,U}$$

in that ordering several times.

Random loopy propagation:
Draw successively links (U, T) uniformly and compute

$$q_{U,T} := p_U \prod_{\substack{W \in \mathrm{fan}(U) \\ W \neq T}} q_{W,U}$$

Random walk loopy propagation:
Draw a start vertex U. Then

(i) draw a vertex $T \in \mathrm{fan}(U)$ and compute

$$q_{U,T} := p_U \prod_{\substack{W \in \mathrm{fan}(U) \\ W \neq T}} q_{W,U}$$

(ii) set $U := T$ and repeat until convergence.

Convergence: computations continue as long as

$$\mathrm{MSE}(\{q'_1, \ldots, q'_n\}, \{q_1, \ldots, q_n\}) > \epsilon$$

with $(q'_i)_{i=1,\ldots,n}$ the last n computed link potentials, q_i the value of link potential q'_i before the last update and ϵ a given threshold for the error (e.g., 0.0001).

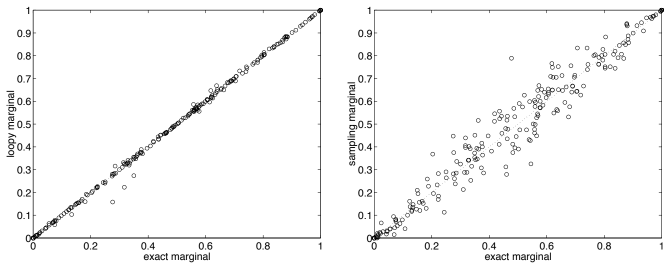


Figure 15: Correlation of true and estimated coefficients using Loopy Propagation ($\epsilon = 10^{-4}$) and LW (200 samples) on PYRAMID network (28 binary variables) [MWJ99, p. 4].

In general, there is no guarantee that loopy propagation converges.

There are example bayesian networks known, for that loopy propagation does not converge (e.g., QMR-DT), but oscillates between different estimates.

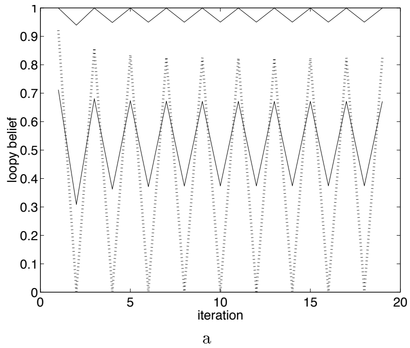


Figure 16: Oscillations of the estimates of three vertices of the QMR-DT network using Loopy Propagation [MWJ99, p. 6].

Loopy propagation has been successfully used in different application areas:

(i) iterative decoding of error-correcting codes (Tanner and factor graphs),

(ii) computer vision (pairwise markov random fields), and

(iii) local magnetizations (Potts and Ising models).

Furthermore there are theoretical underpinnings from statistical physics (Bethe and Kikuchi energy, see [YFW02]) that can help to assess convergence for models with special topologies.

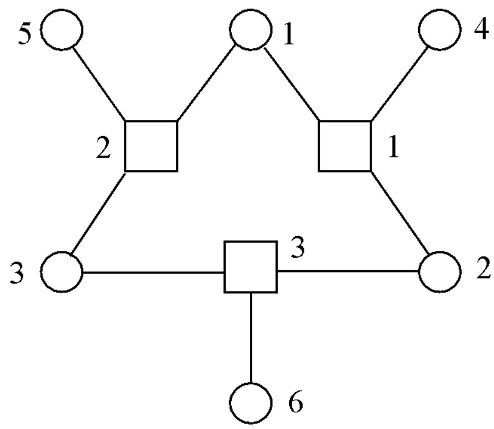


Figure 17: Tanner graph of a 3 bit information in 6 bit messages parity check code [YFW02, p. 6]. Circles denote bits, squares parity checks.

References

[CD00] Jian Cheng and Marek J. Druzdzel. Ais-bn: An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks. *Journal on Artificial Intelligence*, 13:155–188, 2000.

[MWJ99] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on UAI*, 1999.

[SP90] R. D. Shachter and M. Peot. Simulation approaches to general probabilistic inference on belief networks. In M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 221–231. North Holland, Amsterdam, 1990.

[YFW02] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratories, 2002.