Approximate Lifting Techniques for Belief Propagation
Supplementary Material

Parag Singla
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi, 110016, INDIA.
parags@cse.iitd.ac.in

Aniruddh Nath and **Pedro Domingos**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{*nath, pedrod*}*@cs.washington.edu*

1 Message Errors on Factor Graphs

In section 2, we defined μ_{xf} and μ_{fx}, the BP messages, and M_x, the marginal of a variable. We define a similar quantity for factors:

$$M_{fx,i}(\mathbf{x}) = \prod_{y \in nb(f)\setminus\{x\}} \mu_{yf,i}(y_{\mathbf{x}})$$

Let $\hat{\mu}_{xf}$, $\hat{\mu}_{fx}$, $\hat{M}_x$ and $\hat{M}_{fx}$ be our approximations of these quantities. These approximations can be viewed as multiplicative errors on the 'true' quantities at some fixed point of BP:

$$\begin{aligned}
\hat{\mu}_{xf,i}(x) &= \mu_{xf,i}(x) e_{xf,i}(x) \\
\hat{\mu}_{fx,i}(x) &= \mu_{fx,i}(x) e_{fx,i}(x) \\
\hat{M}_{x,i}(x) &= M_{x,i}(x) E_{x,i}(x) \\
\hat{M}_{fx,i}(x) &= M_{fx,i}(x) E_{fx,i}(x)
\end{aligned}$$

(Here, e and E are the multiplicative error functions.)

If the potentials are finite, we can bound the growth of the dynamic range of the error with respect to the operations of BP, using logic very similar to Ihler et al. (2005).

Theorem 2.

$$\begin{aligned}
\log d(e_{xf,i+1}) &\leq \sum_{h \in nb(x)\setminus\{f\}} \log d(e_{hx,i}) \\
\log d(E_{x,i+1}) &\leq \sum_{h \in nb(x)} \log d(e_{hx,i})
\end{aligned}$$

Proof. Both equations can be proved by the same argument as Theorem 6 of Ihler et al. (2005). □

Theorem 3.

$$d(e_{fx,i+1}) \leq \frac{d(f)^2 d(E_{fx,i}) + 1}{d(f)^2 + d(E_{xf,i})}$$

Proof.

$$\begin{aligned}
d(e_{fx,i+1}) &= d(\hat{\mu}_{fx}/\mu_{fx}) \\
&= \max_{a,b} \frac{\sum_{\mathbf{x}_a} f(\mathbf{x}_a) M_{fx}(\mathbf{x}_a) E_{fx}(\mathbf{x}_a)}{\sum_{\mathbf{x}_a} f(\mathbf{x}_a) M_{fx}(\mathbf{x}_a)} \\
&\quad \cdot \frac{\sum_{\mathbf{x}_b} f(\mathbf{x}_b) M_{fx}(\mathbf{x}_b)}{\sum_{\mathbf{x}_b} f(\mathbf{x}_b) M_{fx}(\mathbf{x}_b) E_{fx}(\mathbf{x}_b)}
\end{aligned}$$

The result follows from the same argument as Appendix A of Ihler et al. (2005). □

2 Error Bound for Noisy Hypercubes

The methods of Ihler et al. (2005) can be extended to bound the error introduced by noise-tolerant hypercube formation, using logic similar to Theorem 1 (the error bound for early stopping).

Note that the noisy hypercube approximation is equivalent to flipping the values of certain nodes: true evidence nodes may become false or unknown, or vice versa. For the flipped nodes, the bound is vacuous: the error in the probability may be at most 1. However, we can bound the change in probability on the remaining nodes in the network, by bounding the change in the outgoing messages from the factors.

We can place an upper bound on the errors by assuming that the flipped nodes will maximally alter the outgoing messages from the factors adjacent to them. Since each factor f corresponds to some MLN formula with weight w_f, $f(\mathbf{x})$ for all states is between 1 and e^{w_f}. As a result, the normalized outgoing messages $\mu_{fx}(a)$ to node x are between 1 and e^{w_f} for all a, both before and after the introduction of noise. The dynamic range of the error function can be bounded as follows:

$$d(e_{fx}) \leq \max\left(\sqrt{e^{2w_f}}, \sqrt{e^{-2w_f}}\right)$$

Thus, Theorem 1 can be modified as follows for the noisy hypercube case:

Theorem 4. *If ground BP converges, then for node x, the probability estimated by ground BP at convergence (p_x) can be bounded as follows in terms of the probability $\hat{p}_x$ estimated by lifted BP after n BP steps with some set $X_{flipped}$ of the nodes flipped to a different evidence value.*

$$
\begin{aligned}
p_x &\geq \frac{1}{(\zeta_{x,n})^2[(1/\hat{p}_x) - 1] + 1} = lb(p_x) \\
p_x &\leq \frac{1}{(1/\zeta_{x,n})^2[(1/\hat{p}_x) - 1] + 1} = ub(p_x)
\end{aligned}
$$

where $\log \zeta_{x,n} = \sum_{f \in nb(x)} \log \nu_{fx,n}$,

$$
\log \nu_{xf,i+1} = \sum_{h \in nb(x) \setminus \{f\}} \log \nu_{hx,i}
$$

For factors f adjacent to some node $x \in X_{flipped}$,

$$
\nu_{fx,i} = \max\left(\sqrt{e^{2w_f}}, \sqrt{e^{-2w_f}}\right)
$$

For all other factors f,

$$
\begin{aligned}
\log \nu_{fx,i} &= \log \frac{d(f)^2 \varepsilon_{fx,i} + 1}{d(f)^2 + \varepsilon_{fx,i}} \\
\text{and} \quad \nu_{fx,1} &= d(f)^2 \\
\log \varepsilon_{fx,i} &= \sum_{y \in nb(f) \setminus \{x\}} \log \nu_{yf,i} \\
d(f) &= \sup_{x,y} \sqrt{f(x)/f(y)}
\end{aligned}
$$

3 Additional Experimental Results

3.1 ADDITIONAL DATASETS

3.1.1 Advising Relationships

We predicted advising relationships between students and professors (as described in Richardson and Domingos (2006), using the UW-CSE database and MLN publicly available from the Alchemy website (Kok et al., 2008). We removed the clauses containing existential qualifiers. The database is divided into five areas (AI, graphics, etc.). The database contains a total of 2678 groundings of predicates describing whether someone is a student or professor, who teaches which class, who published which papers, etc. The model was trained using L-BFGS to optimize pseudo-likelihood, using the default parameter settings in Alchemy.

3.1.2 Protein Interactions

We predicted protein interactions in the Yeast Protein dataset from the MIPS (Munich Information Center for Protein Sequence) Comprehensive Yeast Genome Database, as of February 2005 (Mewes et al., 2002). The dataset, originally used in Davis et al. (2005) include information on protein location, function, phenotype, class, and enzymes. It also includes information about protein-protein interactions and protein complexes.

The original data contains information about approximately 4500 proteins and their interactions. We used the processed version this dataset as described by Davis and Domingos (2009). This consists of four disjoint subsamples of the original data, each with around 450 proteins. To create each subsample, starting with a randomly selected seed set of proteins, all previously unselected proteins that appeared within two links (via the interaction predicate) of the seed set were included. The goal was predict the interaction relation. We used the MLN learned by the Refine algorithm described by Davis and Domingos (2009).

References

Davis, J., Burnside, E., Dutra, I., Page, D., & Costa, V. S. (2005). An integrated approach to learning Bayesian networks of rules. *Proc. of ECML-05*.

Davis, J., & Domingos, P. (2009). Deep transfer via second-order Markov logic. *Proc. of ICML-09*.

Ihler, A. T., Fisher, J. W., & Willsky, A. S. (2005). Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, *6*.

Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., & Domingos, P. (2008). *The Alchemy system for statistical relational AI* (Technical Report). University of Washington. http://alchemy.cs.washington.edu.

Mewes, H. W., Frishman, D., Güldener, U., Mannhaupt, G., Mayer, K., Mokreis, M., Morgenstern, B., Münsterkötter, M., Rudd, S., & Weil, B. (2002). MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, *30*.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*.

Table 1: Experimental Results. Memory is in MB; Features and Tuples are in thousands.

	Algorithm	**Time (in seconds)**			**Memory**			**Accuracy**	
		Construct	BP	Total	Memory	Features	Tuples	CLL	AUC
UW-CSE	Ground	**1.6**	502.0	503.7	101	227.3	227.3	**-0.022**	**0.338**
	Extensional	7.9	215.7	223.6	193	92.1	227.3	**-0.022**	**0.338**
	Resolution	8.0	214.8	222.9	193	92.1	227.3	**-0.022**	**0.338**
	Hypercube	19.2	232.9	252.1	180	92.1	92.4	**-0.022**	**0.338**
	Early Stop	4.1	100.5	104.6	80	47.6	86.1	**-0.022**	**0.338**
	Noise-Tol.	8.1	**91.6**	**99.8**	**76**	**37.0**	**37.4**	-0.024	0.224
Yeast	Ground	**34.8**	1743.0	1777.9	426	639.5	639.5	**-0.033**	**0.043**
	Extensional	75.4	5.9	**81.3**	443	142.4	639.5	**-0.033**	**0.043**
	Resolution	77.9	6.1	84.0	443	142.4	639.5	**-0.033**	**0.043**
	Hypercube	206.8	1.9	208.7	354	142.4	146.4	**-0.033**	**0.043**
	Early Stop	207.4	1.9	209.3	354	142.4	146.4	**-0.033**	**0.043**
	Noise-Tol.	97.7	**1.3**	99.0	**304**	**107.8**	**100.1**	**-0.033**	**0.043**

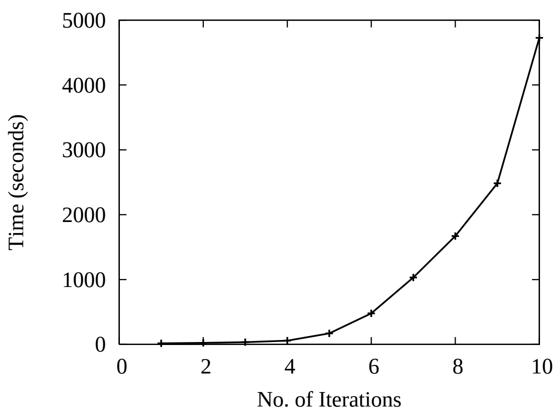

Figure 1: Time vs. number of iterations for early stopping on Denoise.

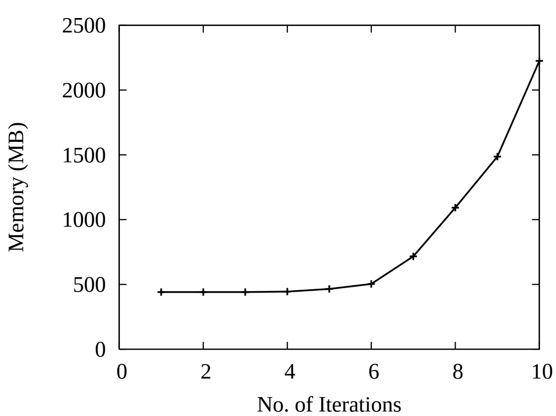

Figure 2: Memory vs. number of iterations for early stopping on Denoise.

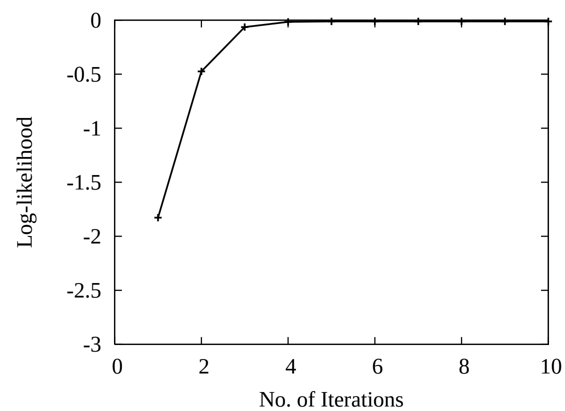

Figure 3: Log-likelihood vs. number of iterations for early stopping on Denoise.

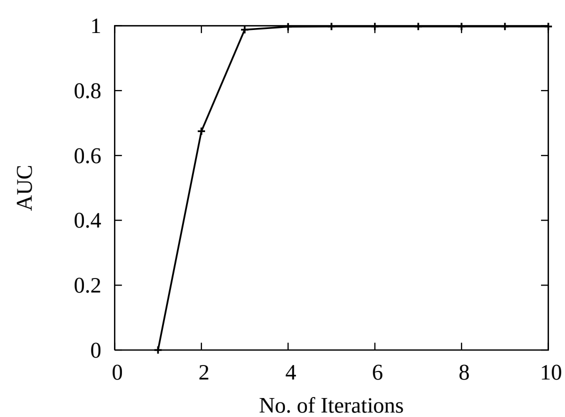

Figure 4: AUC vs. number of iterations for early stopping on Denoise.

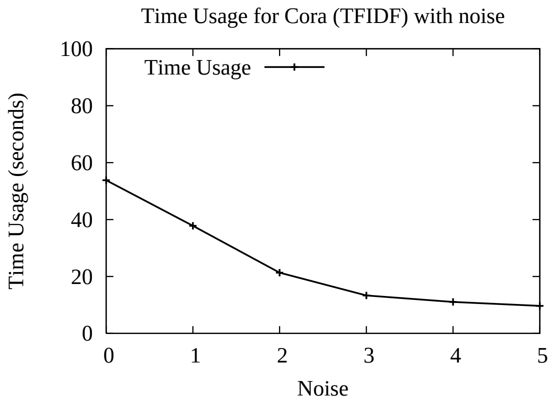

Figure 5: Time vs. noise tolerance on Cora.

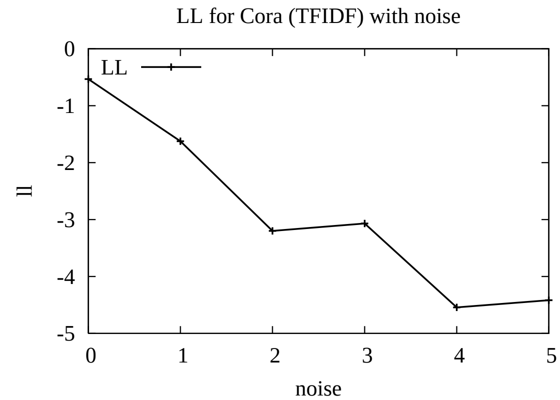

Figure 7: Log-likelihood vs. noise tolerance on Cora.

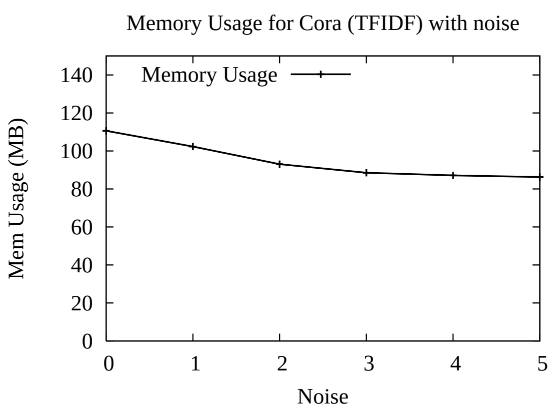

Figure 6: Memory vs. noise tolerance on Cora.

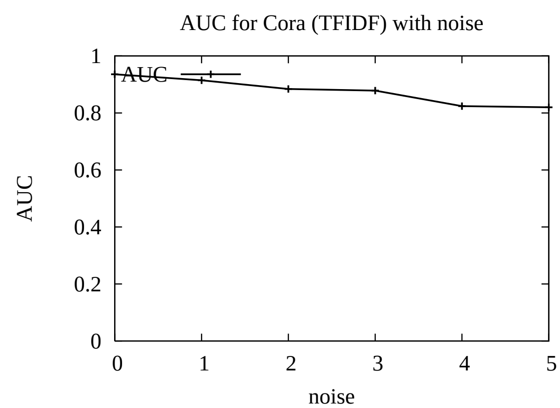

Figure 8: AUC vs. noise tolerance on Cora.