Amortized Bethe Free Energy Minimization for Learning MRFs

Sam Wiseman and Yoon Kim

Motivating Questions

- How good are popular approximate inference methods at learning (deep) structured models with discrete latent variables?
- Are there learning objectives that don't require sampling-based gradient estimators?

Main Idea

- **TL;DR:** Learn MRFs by minimizing what loopy belief propagation (LBP) does, but faster, with inference networks rather than message passing.
- Use the Bethe free energy (BFE) partition function approximation; requires no sampling.
- This is only advantageous for undirected models.

Bethe Approximations

Notation:

- Let $\mathcal{G} = (\mathcal{V} \cup \mathcal{F}, \mathcal{E})$ be a factor graph, with $\mathbf{x} \subseteq \mathcal{V}$ observed and $\mathbf{z} \subseteq \mathcal{V}$ latent.
- Let Ψ_α be potential associated with factor α and $\mathbf{x}_\alpha, \mathbf{z}_\alpha$ be participating subvectors.
- $Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}'} \sum_{\mathbf{z}'} \prod_\alpha \Psi_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha; \boldsymbol{\theta})$.
- $Z(\mathbf{x}, \boldsymbol{\theta}) = \sum_{\mathbf{z}'} \prod_\alpha \Psi_\alpha(\mathbf{x}_\alpha, \mathbf{z}'_\alpha; \boldsymbol{\theta})$.

BFE (Bethe, 1935; Yedidia et al., 2001):

$$F(\boldsymbol{\tau}, \boldsymbol{\theta}) = \mathrm{KL}[Q_{\boldsymbol{\tau}} || P(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})] - \log Z(\boldsymbol{\theta})$$
$$= \sum_\alpha \sum_{\mathbf{x}'_\alpha, \mathbf{z}'_\alpha} \tau_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha) \log \frac{\tau_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha)}{\Psi_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha)}$$
$$- \sum_{v \in \mathcal{V}} (|\mathrm{ne}(v)| - 1) \sum_{v'} \tau_v(v') \log \tau_v(v')$$

- Let $\boldsymbol{\tau}_\alpha(\mathbf{x}_\alpha, \mathbf{z}_\alpha)$ be Ψ_α's (pseudo) marginals, and $\mathcal{C}$ contain all locally consistent assignments $\forall \alpha$.
- For a tree, $\min_{\boldsymbol{\tau} \in \mathcal{C}} F(\boldsymbol{\tau}, \boldsymbol{\theta}) = -\log Z(\boldsymbol{\theta})$.
- Otherwise, $\min_{\boldsymbol{\tau} \in \mathcal{C}} F(\boldsymbol{\tau}, \boldsymbol{\theta}) \approx -\log Z(\boldsymbol{\theta})$.
- Loopy BP finds stationary points of $F(\boldsymbol{\tau}, \boldsymbol{\theta})$ (Yedidia et al., 2001).

Why the BFE is Attractive

- Only linear in the number of factors!
- But, having many low-degree factors is only interesting for MRFs (c.f., products of experts (Hinton, 2002)).

A BFE-based Objective

- Replace partition functions in the log marginal with their BFE approximations:
$$\log \tilde{P}(\mathbf{x}; \boldsymbol{\theta}) + \min_{\boldsymbol{\tau} \in \mathcal{C}} F(\boldsymbol{\tau}) \approx \log \tilde{P}(\mathbf{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})$$
- Gives rise to a saddle-point objective:
$$\min_{\boldsymbol{\theta}} \left[-\log \tilde{P}(\mathbf{x}; \boldsymbol{\theta}) - \min_{\boldsymbol{\tau} \in \mathcal{C}} F(\boldsymbol{\tau}, \boldsymbol{\theta}) \right]$$
$$= \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\tau} \in \mathcal{C}} \left[-\log \tilde{P}(\mathbf{x}; \boldsymbol{\theta}) - F(\boldsymbol{\tau}, \boldsymbol{\theta}) \right]$$
- If there are latents:
$$\min_{\boldsymbol{\theta}} \left[\min_{\boldsymbol{\tau}_{\mathbf{x}} \in \mathcal{C}_{\mathbf{x}}} F(\boldsymbol{\tau}_{\mathbf{x}}, \boldsymbol{\theta}) - \min_{\boldsymbol{\tau} \in \mathcal{C}} F(\boldsymbol{\tau}, \boldsymbol{\theta}) \right]$$
$$= \min_{\boldsymbol{\theta}, \boldsymbol{\tau}_{\mathbf{x}}} \max_{\boldsymbol{\tau} \in \mathcal{C}} \left[F(\boldsymbol{\tau}_{\mathbf{x}}, \boldsymbol{\theta}) - F(\boldsymbol{\tau}, \boldsymbol{\theta}) \right]$$

Amortized Inference

- Train inference networks $f(\cdot; \boldsymbol{\phi})$, $f_{\mathbf{x}}(\cdot; \boldsymbol{\phi}_{\mathbf{x}})$ to approximately minimize $F(\boldsymbol{\tau}, \boldsymbol{\theta})$, $F(\boldsymbol{\tau}_{\mathbf{x}}, \boldsymbol{\theta})$.
- But predicted pseudo-marginals must normalize and be locally consistent.
- Define $\boldsymbol{\tau}_\alpha(\mathbf{x}_\alpha, \mathbf{z}_\alpha; \boldsymbol{\phi}) = \mathrm{softmax}(\mathbf{f}(\mathcal{G}, \alpha; \boldsymbol{\phi}))$.
- Obtain predicted node-marginals as:
$$\boldsymbol{\tau}_v(v; \boldsymbol{\phi}) = \frac{1}{|\mathrm{ne}(v)|} \sum_{\alpha \in \mathrm{ne}(v)} \sum_{\mathbf{x}'_\alpha, \mathbf{z}'_\alpha \setminus v} \boldsymbol{\tau}_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha; \boldsymbol{\phi})$$
- Handle local consistency by penalizing deviation from $\boldsymbol{\tau}_v(v; \boldsymbol{\phi})$.
- Final objective:
$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \Big[-\log \tilde{P}(\mathbf{x}; \boldsymbol{\theta}) - F(\boldsymbol{\tau}(\boldsymbol{\phi}), \boldsymbol{\theta}) \qquad (1)$$
$$- \lambda \sum_{\substack{v \in \mathcal{V} \\ \alpha \in \mathrm{ne}(v)}} d\big(\boldsymbol{\tau}_v(v; \boldsymbol{\phi}), \sum_{\mathbf{x}'_\alpha, \mathbf{z}'_\alpha \setminus v} \boldsymbol{\tau}_\alpha(\mathbf{x}'_\alpha, \mathbf{z}'_\alpha; \boldsymbol{\phi})\big) \Big]$$
- If there are latents, replace $-\log \tilde{P}$ with $F(\boldsymbol{\tau}_{\mathbf{x}}, \boldsymbol{\theta})$ and add additional penalty terms for $\boldsymbol{\phi}_{\mathbf{x}}$.

Learning

Alternating Gradient Ascent/Descent:

- Take I_1 gradient ascent steps on (1) wrt $\boldsymbol{\phi}$.
- If there are latents, take I_2 gradient descent steps on (1) wrt $\boldsymbol{\phi}_{\mathbf{x}}$.
- Take a gradient descent step on (1) wrt $\boldsymbol{\theta}$.

Ising Models

Just inference:

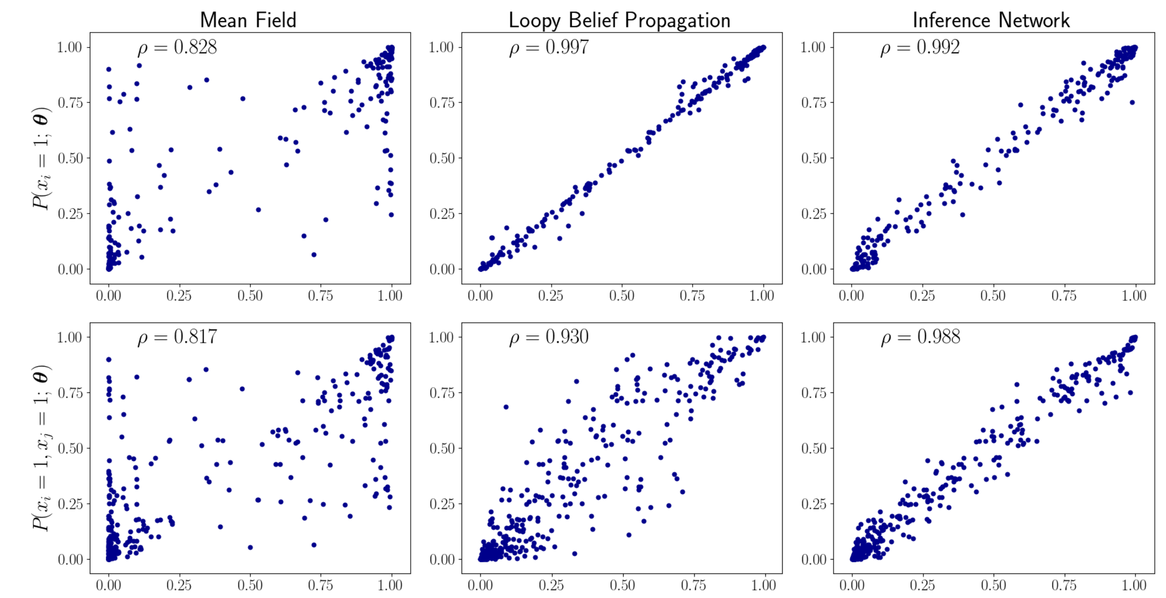

Figure 1 Approximate marginals (x-axis) against the true marginals (y-axis) for a 15 × 15 Ising model. Top: node marginals; bottom: pairwise factor marginals.

Learning:

n	True Ent.	Rand. Init	Exact	Mean Field	LBP	Inf. Net
5	6.27	45.62	6.30	7.35	7.17	6.47
10	25.76	162.53	25.89	29.70	28.34	26.80
15	51.80	365.36	52.24	60.03	59.79	54.91

Table 1 Held out NLL. 'True Ent.' is NLL under the true model (i.e. $\mathbb{E}_{P(\mathbf{x};\boldsymbol{\theta})}[-\log P(\mathbf{x};\boldsymbol{\theta})]$), and 'Exact' trains with the exact partition function. The Inf. Net is a 1-layer Transformer (Vaswani et al., 2017).

Restricted Boltzmann Machines

- Following Kuleshov & Ermon (2017), we train RBMs with 100 hidden units on the UCI digits dataset.
- We compare with persistent contrastive divergence (Tieleman, 2008), LBP (10 random sweeps), and the variational approach of Kuleshov & Ermon (2017).
- Our inference network runs a bidirectional LSTM (Hochreiter & Schmidhuber, 1997) over the linearized graph.

	NLL	ℓ_F	Speedup
Loopy BP	25.47	53.02	1
Inference Network	23.43	23.11	1544x
PCD	21.24	N/A	21617x
Kuleshov & Ermon (2017)	$\geq$ 24.5		

Table 2 Held out average NLL of RBMs, as estimated by AIS (Salakhutdinov & Murray, 2008).

High-order HMMs

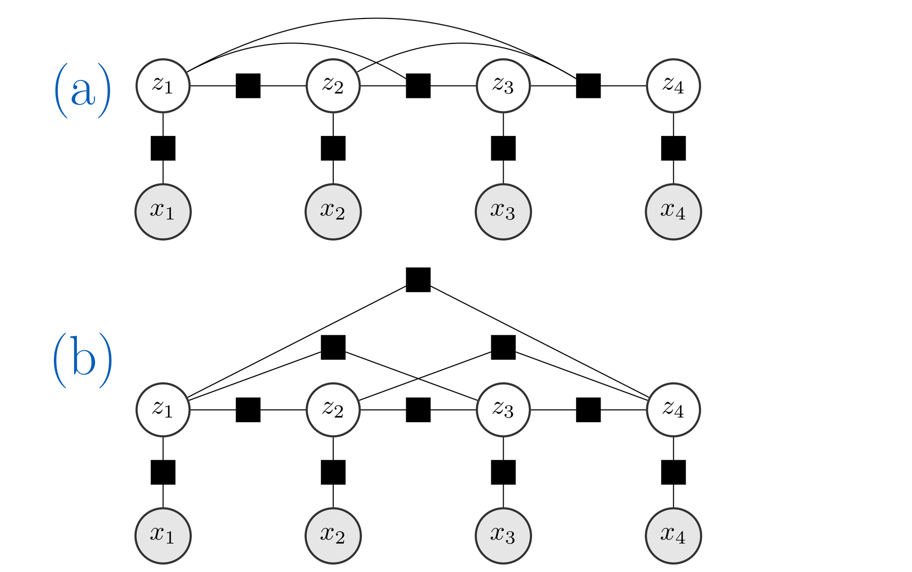

Figure 2 Top: standard 3rd order HMM; bottom: pairwise, product-of-expert MRF HMM.

Why?

- Approximate inference techniques can be evaluated exactly.
- Natural to define an undirected HMM analog.

Experiments:

- 3rd order neural HMM (Tran et al., 2016), on Penn Treebank sentences, $K = 30$.
- We compare average NLL of exact inference, discrete VAE variants, LBP and amortized BFE minimization.

Directed/VAE models:

- Neural HMM: emission and transition distributions parameterized by feed-fwd nets.
- Mean-field (MF) inf. net: BLSTM over input into linear decoder for each token.
- First-order (FO) inf. net: 1st order neural HMM; conditions on averaged BLSTM states of input.

MRF/Bethe models:

- Pairwise MRF HMM: transition factors are feed-fwd function of distance; emissions as above.
- Bethe inf. net: BLSTM over embeddings of MRF nodes into linear to predict marginals.

Results:

	NLL	-ELBO/$\ell_{F,\mathbf{z}}$	Speedup
Exact	105.66	105.66	1.00
Mean-Field VAE + BL	119.27	175.46	1.67
Mean-Field IWAE-10	119.20	167.71	0.16
1st Order HMM VAE	118.35	118.88	0.73
Exact	104.07	104.07	1.12
LBP	108.74	99.89	0.55
Inference Network	115.86	114.75	1.96

Table 3 Top: directed HMM models; bottom: undirected, pairwise HMM variant.