Inference in Probabilistic Graphical Models by Graph Neural Networks

KiJung Yoon, Renjie Liao, Yuwen Xiong, Lisa Zhang, Ethan Fetaya, Raquel Urtasun, Richard Zemel, Xaq Pitkow

Presenter: Arshdeep Sekhon

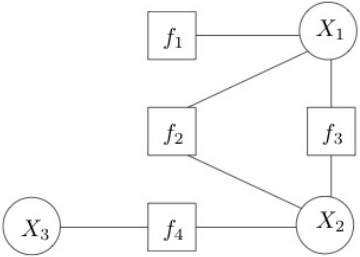
Outline

Probabilistic Graphical Models(PGMs)

- Given $\mathbf{x} \in R^D$, joint probability $p(\mathbf{x})$
- Simplify joint $p(\mathbf{x})$ into a factorization based on conditional independence defined by a graph structure.

Factor Graphs

- Factorization of the joint probability distribution for more efficient computations
- bipartite graph: two types of nodes, edges connect different node types
- Given a factorization:
  $g(X_1, X_2, X_3) = f_1(X_1) f_2(X_1, X_2) f_3(X_1, X_2) f_4(X_2, X_3)$



Tasks for PGMs: Inference

Inference Task: Given a graphical model $p(\mathbf{x})$, find marginal probability $p_i(x_i)$
$p_i(x_i) = \sum_{x/x_i} p(\mathbf{x})$

Tasks: MAP

Maximum A Posteriori(MAP) Inference: $\boldsymbol{x}* = argmax_x p(\boldsymbol{x})$, Finding the most probable state

Belief Propagation

Belief propagation operates on these factor graphs by constructing messages $\mu_{i \to \alpha}$ and $\mu_{\alpha \to i}$ that are passed between variable(i) and factor(α) nodes:

$$\mu_{\alpha \to i}(x_i) = \sum_{\mathbf{x}_\alpha \setminus x_i} \psi_\alpha(\mathbf{x}_\alpha) \prod_{j \in N_\alpha \setminus i} \mu_{j \to \alpha}(x_j) \tag{1}$$

$$\mu_{i \to \alpha}(x_i) = \prod_{\beta \in N_i \setminus \alpha} \mu_{\beta \to i}(x_i) \tag{2}$$

the estimated marginal joint probability of a factor α, namely $B_\alpha(\mathbf{x}_\alpha)$, is given by

$$B_\alpha(\mathbf{x}_\alpha) = \frac{1}{Z} \psi_\alpha(\mathbf{x}_\alpha) \prod_{i \in N\alpha} \mu_{i \to \alpha}(x_i) \tag{3}$$

Belief Propagation

Issues:

- Exact Inference on tree graphs, but not on graphs with cycles
- Update Steps may not have closed form solutions

Special Case: Binary Markov Random Field

- variables $x \in \{+1, -1\}^{|V|}$

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(\mathbf{b} \cdot \mathbf{x} + \mathbf{x} \cdot J \cdot \mathbf{x}\right) \tag{4}$$

- singleton factor: $\psi_i(x_i) = e^{b_i x_i}$
- pairwise factors: $\psi_{i,j}(x_i, x_j) = e^{J_{ij} x_i x_j}$
- Goal: find $p(x_i)$
- $\mathbf{J}$ is a symmetric matrix

Belief Propagation on Binary Markov Fields

Belief propagation updates messages μ_{ij} from i to j according to

$$\mu_{ij}(x_j) = \sum_{x_i} e^{J_{ij}x_i x_j + b_i x_i} \prod_{k \in N_i \backslash j} \mu_{ki}(x_i) \tag{5}$$

estimated marginals by $\hat{p}_i(x_i) = \frac{1}{Z} e^{b_i x_i} \prod_{k \in N_i} \mu_{ki}(x_i)$

Proposed GNN architecture

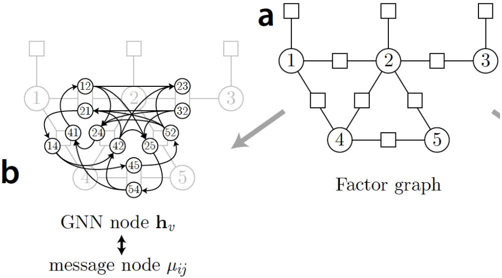
$$\mathbf{m}_{i\to j}^{t+1} = \mathcal{M}(\mathbf{h}_i^t, \mathbf{h}_j^t, \varepsilon_{ij}) \tag{6}$$

$$\mathbf{m}_i^{t+1} = \sum_{j \in N_i} \mathbf{m}_{j\to i}^{t+1} \tag{7}$$

$$\mathbf{h}_i^{t+1} = \mathcal{U}(\mathbf{h}_i^t, \mathbf{m}_i^{t+1}) \tag{8}$$

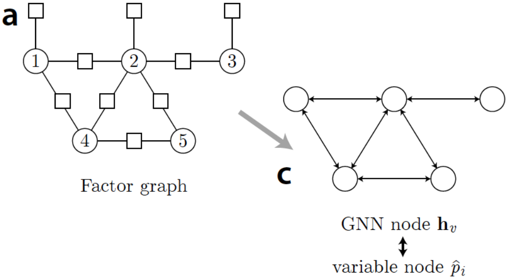
$$\hat{\mathbf{y}} = \sigma\left(\mathcal{R}(\mathbf{h}_i^{(T)})\right) \tag{9}$$

Proposed Model: Message-GNN



- Convert all messages $\mu_{i\to j}$ into a node in a GNN $\boldsymbol{h}_{i\to j}$
- Two GNN nodes v and w are connected if they correspond to messages $\mu_{i\to j}$ and $\mu_{j\to k}$
- message from v_i to v_j is computed by $\mathbf{m}^{t+1}_{i\to j} = \mathcal{M}(\sum_{k\in N_i \setminus j} \mathbf{h}^t_{k\to i}, e_{ij})$.
- update its hidden state by $\mathbf{h}^{t+1}_{i\to j} = \mathcal{U}(\mathbf{h}^t_{i\to j}, \mathbf{m}^{t+1}_{i\to j})$.
- Readout: $\hat{y}(x_i) = \mathcal{R}(\sum \ldots \mathbf{h}^{(T)} \ldots)$

Proposed Model: node-GNN



- No representation for factor nodes
- information about interactions in ϵ_{ij}

GNN for inference and MAP

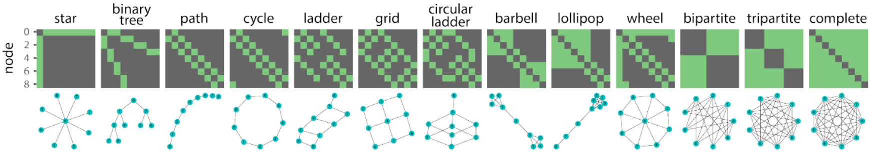
- minimize cross entropy loss $L(p, \tilde{p}) = -\sum_i q_i log \hat{p}_i(x_i)$
- For MAP: delta function $q_i = \delta_{x_i, x_i^*}$
- For Marginal Inference: q_i enumeration of ground truth

Experimental Design

- generalization under 4 conditions
- to unseen graphs of the same structure (I, II),
- and to completely different random graphs (III, IV).
- These graphs may be the same size (I, III) or larger (II, IV).

	structured	random
$n = 9$	I	III
$n = 16$	II	IV

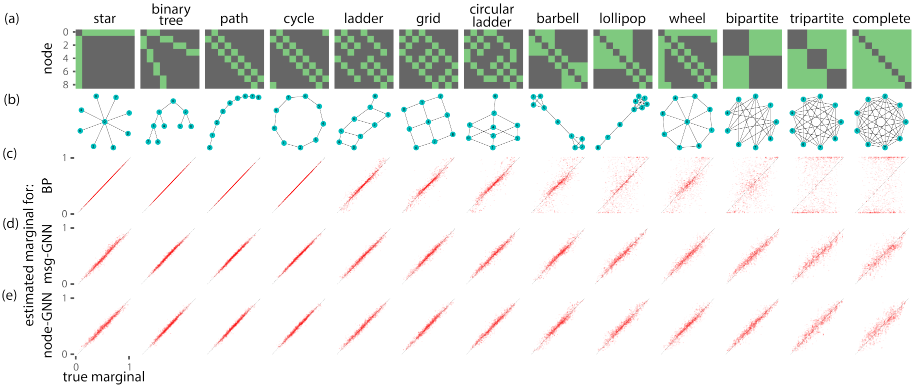
Experimental Set up



- train on 100 graphical models of 13 classical types
- Sample $J_{ij} = J_{ji} \sim \mathcal{N}(0, 1)$
- sample biases $b_i \sim \mathcal{N}(0, (1/4)^2)$

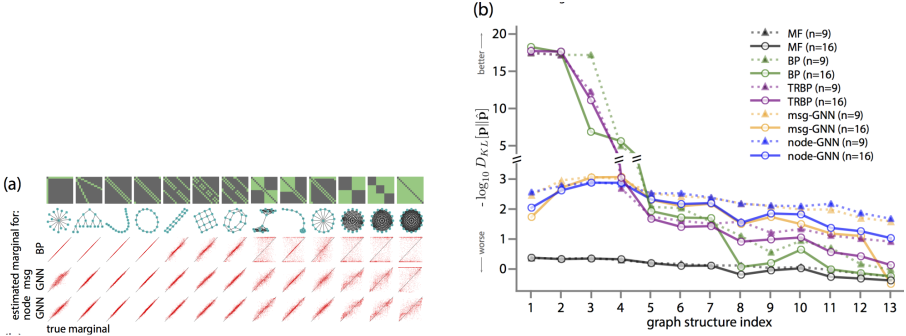
Within Set Generalization

- test graphs had the same size and structure as training graphs
- but the values of singleton and edge potentials differed
- most notable performance difference between loopy graphs



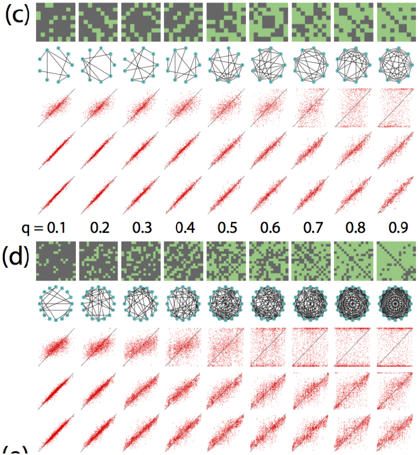
Out of Set Generalization

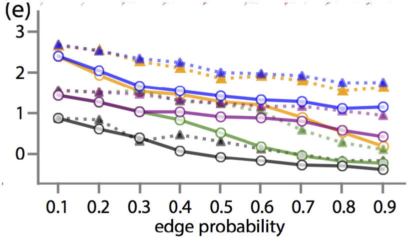
- Train on same graphs
- Test on bigger graphs
- Metric: the average Kullback-Leibler divergence $\langle D_{KL}[p_i(x_i) \| \hat{p}_i(x_i)] \rangle$ across the entire set of test graphs with the small and large number of nodes.



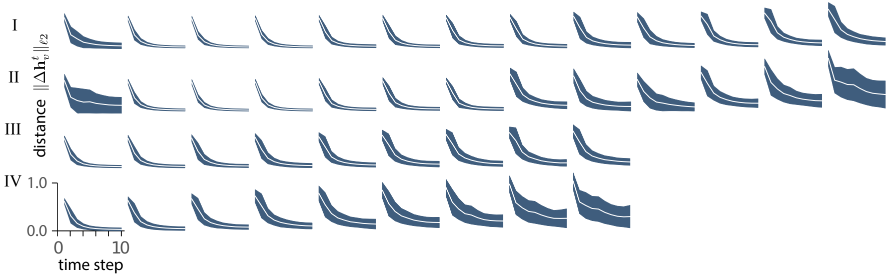
Out of Set Generalization: different structure

- connected random Erdos Renyi graphs $G_{n,q}$,
- changed connectivity by increasing the edge probability from q $= 0.1$ (sparse) to 0.9 (dense)



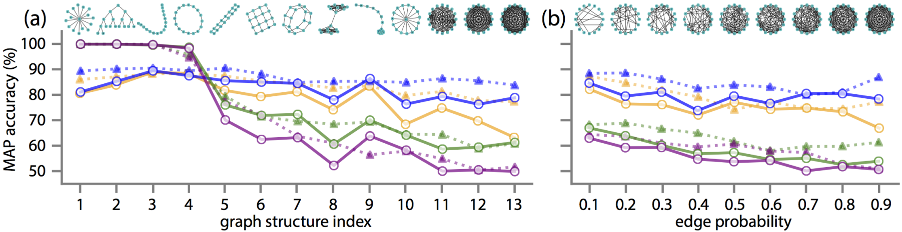


- How node states change over time
- $||h_v^t - h_v^{t-1}||_{\ell_2}$



MAP Estimation

- $x* = argmax_x p(\mathbf{x})$



Conclusions

- limited testing: binary markov random field models only
- relatively small graphs
- A combination of NNs approximation power to incorporate non linear structure of inference problems