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

1. Introduction
2. Proposed Method
3. Experiments
Given $\mathbf{x} \in \mathbb{R}^D$, 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:
  \[ g(X_1, X_2, X_3) = f_1(X_1)f_2(X_1, X_2)f_3(X_1, X_2)f_4(X_2, X_3) \]
Inference Task: Given a graphical model $p(x)$, find marginal probability

$$p_i(x_i) = \sum_{x/x_i} p(x)$$
Maximum A Posteriori (MAP) Inference: $x^* = \arg\max_x p(x)$, Finding the most probable state
Belief Propagation

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

$$
\mu_{\alpha \rightarrow i}(x_i) = \sum_{x_\alpha \setminus x_i} \psi_\alpha(x_\alpha) \prod_{j \in N_\alpha \setminus i} \mu_{j \rightarrow \alpha}(x_j) \quad (1)
$$

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

the estimated marginal joint probability of a factor $\alpha$, namely $B_\alpha(x_\alpha)$, is given by

$$
B_\alpha(x_\alpha) = \frac{1}{Z} \psi_\alpha(x_\alpha) \prod_{i \in N_\alpha} \mu_{i \rightarrow \alpha}(x_i) \quad (3)
$$
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 $x \in \{+1, -1\}^{|V|}$

$$p(x) = \frac{1}{Z} \exp (b \cdot x + x \cdot J \cdot x)$$  \hspace{1cm} (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 $p(x_i)$
- $J$ is a symmetric matrix
Belief Propagation updates messages $\mu_{ij}$ from $i$ to $j$ according to

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

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

\[ m_{i \rightarrow j}^{t+1} = \mathcal{M}(h_i^t, h_j^t, \varepsilon_{ij}) \]  

(6)

\[ m_i^{t+1} = \sum_{j \in N_i} m_{j \rightarrow i}^{t+1} \]  

(7)

\[ h_i^{t+1} = \mathcal{U}(h_i^t, m_i^{t+1}) \]  

(8)

\[ \hat{y} = \sigma \left( \mathcal{R}(h_i^{T}) \right) \]  

(9)
Proposed Model: Message-GNN

- Convert all messages $\mu_{i \rightarrow j}$ into a node in a GNN $h_{i \rightarrow j}$
- Two GNN nodes $v$ and $w$ are connected if they correspond to messages $\mu_{i \rightarrow j}$ and $\mu_{j \rightarrow k}$
- Message from $v_i$ to $v_j$ is computed by $m_{i \rightarrow j}^{t+1} = M(\sum_{k \in N_i \setminus j} h_{k \rightarrow i}^t, e_{ij})$.
- Update its hidden state by $h_{i \rightarrow j}^{t+1} = U(h_{i \rightarrow j}^t, m_{i \rightarrow j}^{t+1})$.
- Readout: $\hat{p}(x_i) = R(\sum_i h_{i \rightarrow j}(T))$. 

GNN node $h_v$
message node $\mu_{ij}$

Factor graph
Proposed Model: node-GNN

- No representation for factor nodes
- Information about interactions in $\epsilon_{ij}$
GNN for inference and MAP

- minimize cross entropy loss $L(p, \hat{p}) = -\sum_i q_i \log \hat{p}_i(x_i)$
- 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).

<table>
<thead>
<tr>
<th>structured</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 9$</td>
<td>I</td>
</tr>
<tr>
<td>$n = 16$</td>
<td>II</td>
</tr>
</tbody>
</table>
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)
Convergence of Inference Dynamics

- How node states change over time
- \( \| h_v^t - h_v^{t-1} \|_2 \)
\[ x^* = \arg\max_x p(x) \]
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