Logical Distillation of GNNs

Alexander Pluska, Pascal Welke, Thomas Gärtner, Sagar Malhotra

Main Contributions

- We introduce $\mathcal{EMLC}\%$, a logic closely related to C^2 which has strong connections to GNNs.
- We present *Iterated Decision Trees* (IDTs), a novel method for distilling $\mathcal{EMLC}\%$ formulas from GNNs.
- The distilled IDTs often surpass the accuracy of the underlying GNN while providing insight into their decision process.

$\mathcal{EMLC}\%$

Definition 1 *A* modal parameter S *is one of the following*

$$0, 1, I, A, 1-I, 1-A, I+A, 1-I-A.$$

An $\mathcal{EMLC}\%$ formula is then built by the grammar

$$\varphi ::= U \mid \top \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \neg\varphi \mid S\varphi > n \mid S\varphi > p$$

- *U ranges over node features,*
- *S ranges over modal parameters,*
- *n ranges over $\mathbb{N}$ and p ranges over the interval $(0,1)$.*

Example 1 *Each $\mathcal{EMLC}\%$ formula expresses a node property. The modal parameters are used to express properties of other nodes, e.g.*

$$A({\color{blue}\bullet} \lor {\color{orange}\bullet}) > 2$$

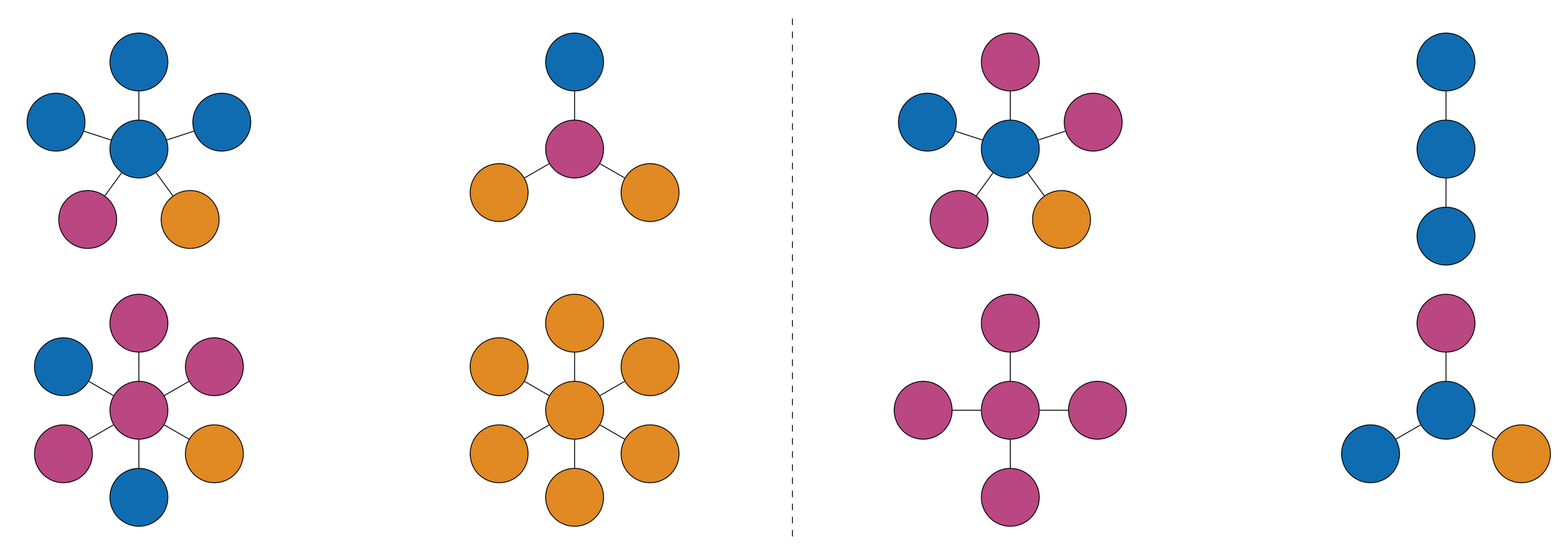
expresses that a node has more than 2 neighbors that are blue or orange,

$$(I+A){\color{magenta}\bullet} > 1/3$$

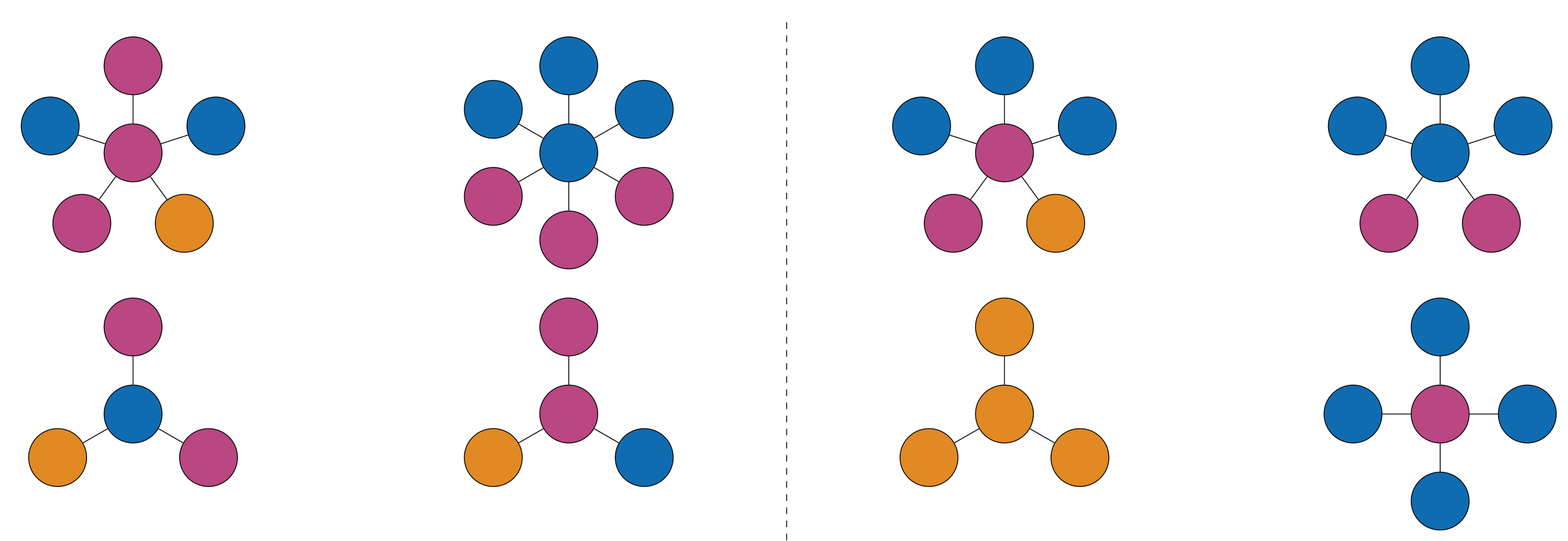
expresses that among the node and its neighbors, more than a third are red and

$$1{\color{blue}\bullet} > 10$$

expresses that more than 10 nodes in the graph are blue.



Examples (left) and non-examples (right) of nodes satisfying the formula $A({\color{blue}\bullet} \lor {\color{orange}\bullet}) > 2$.



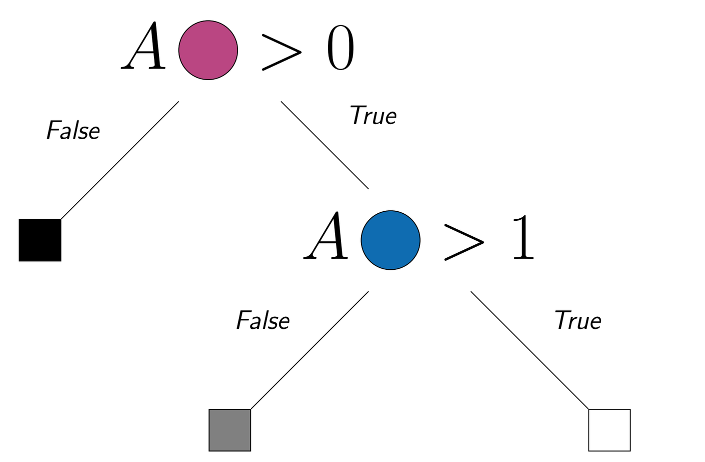
Examples (left) and non-examples (right) of nodes satisfying the formula $(I+A){\color{magenta}\bullet} > 1/3$.

Iterated Decision Trees

Definition 2 *An iterated decision tree layer L consists of*

- *a decision tree T with splitting decisions of the form $S\varphi > x$,*
- *a set of leaf sets of T.*

Example 2 *Consider the following iterated decision tree layer*



with a set of leaf sets $\{M, N\}$ where $M = \{\blacksquare\}$, $N = \{\blacksquare, \square\}$. These leaf sets correspond to node properties

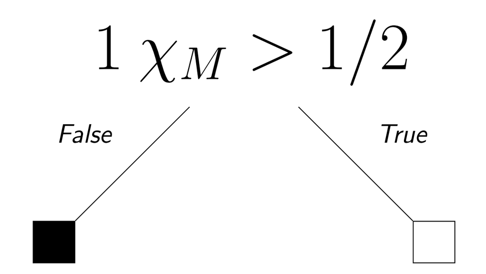
$$\chi_M \iff \neg(A{\color{magenta}\bullet} > 0)$$
$$\chi_N \iff \neg(A{\color{magenta}\bullet} > 0) \lor (A{\color{blue}\bullet} > 1).$$

which express that a node has no red neighbors and that a node has no red neighbors or at least two blue neighbors, respectively.

Definition 3 *An* iterated decision tree *is a sequence of iterated decision tree layers, such that the splitting decisions of each layer are expressed in terms of formulas which*

- *are node features of the input graph (e.g. red),*
- *correspond to leaf sets of previous layers (e.g. χ_M).*

Example 3 *The layer from Example 2 can be extended by*



with a single leaf set $O = \{\square\}$, which corresponds to

$$\chi_O \iff 1\chi_M > 1/2 \iff 1(\neg(A{\color{magenta}\bullet} > 0)) > 1/2$$

expressing "more than half of the nodes have no red neighbor".

Learning IDTs

IDTs are learned from GNNs by iteratively training IDT layers on the intermediate representations of the GNN of the training set $\mathcal{G}$. We use $\mathcal{X}^{(k)}$ to denote the set of all node representations computed after k iterations of the GNN. Note that $\mathcal{X}^{(0)}$ is simply the set of initial node features.

Algorithm 1 Learning Procedure for Iterated Decision Trees

```
procedure LearnIDT(𝒢, 𝒳_GNN)
    IDT ← ∅
    𝒰 ← 𝒳^(0)
    for k ∈ [l + 1] do
        L ← LearnIDTLayer(𝒢, 𝒰, 𝒳^(k+1))
        for M ∈ LeafSets(L) do
            𝒰 ← Append(𝒰, FeatureVector(𝒢, χ_M))
        end for
        IDT ← Append(IDT, L)
    end for
    return IDT
end procedure
```