End-to-End Learning of Probabilistic Hierarchies on Graphs

Daniel Zügner, Bertrand Charpentier, Morgane Ayle,
Sascha Geringer, Stephan Günnemann

ICLR 2022
Technical University of Munich
Real-World Graphs are Often Hierarchical

- Real-world graphs are often hierarchically organized.
- E.g., in citation networks, subgraphs are more densely connected the more specialized they become.
- Hierarchical clustering aims to discover the latent hierarchy in the data.

Figure: [Girvan and Newman 2002]
Problem Outline

• Since we typically do not have ground-truth information about the underlying hierarchy, we mostly rely on unsupervised learning with internal metrics.

• Existing quality metrics are discrete, i.e., not differentiable.

• Existing clustering approaches are mostly heuristic and not directly related to the evaluation metrics.
Contributions

1. We propose a **probabilistic model over hierarchies** via **continuous relaxation** of a tree’s parent assignment matrices.

2. We **theoretically analyze** the model by drawing connections to **absorbing Markov chains**, which

3. allows **efficient and exact computation** of lowest-common-ancestor (LCA) probabilities, which enables us to

4. learn **hierarchies on graphs** by **end-to-end optimization** of relaxed versions of quality metrics such as **Dasgupta cost** and **TSD score**.
Probabilistic Hierarchy Model

• A **continuous relaxation** of discrete tree hierarchies.

• Learnable matrices $A$ and $B$ contain **parent assignment probabilities** for leaves and internal nodes.

• We can easily obtain **valid tree hierarchies** via row-wise sampling from $A$ and $B$. 
Probabilistic Hierarchy Model

• A continuous relaxation of discrete tree hierarchies.

• Learnable matrices $A$ and $B$ contain parent assignment probabilities for leaves and internal nodes.

• We can easily obtain valid tree hierarchies via row-wise sampling from $A$ and $B$. 
Probabilistic Hierarchy Model

• A **continuous relaxation** of discrete tree hierarchies.

• Learnable matrices $A$ and $B$ contain **parent assignment probabilities** for leaves and internal nodes.

• We can easily obtain **valid tree hierarchies** via row-wise sampling from $A$ and $B$. 

\[
p(z_j \mid z_i) \quad \text{Learned Hierarchy}
\]

\[
p(z_j \mid v_i) \quad \text{Observed Graph (leaf nodes)}
\]
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete tree hierarchies.
- Learnable matrices $A$ and $B$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $A$ and $B$. 
• **A continuous relaxation** of discrete tree hierarchies.

• Learnable matrices $A$ and $B$ contain **parent assignment probabilities** for leaves and internal nodes.

• We can easily obtain **valid tree hierarchies** via row-wise sampling from $A$ and $B$. 

---

### Observed Graph
- (leaf nodes)

### Learned Hierarchy

### Matrix Form
- $A$
- $B$
Probabilistic Hierarchy Model

• A continuous relaxation of discrete tree hierarchies.
• Learnable matrices $A$ and $B$ contain parent assignment probabilities for leaves and internal nodes.
• We can easily obtain valid tree hierarchies via row-wise sampling from $A$ and $B$. 
Hierarchical Clustering Metrics

Two established internal metrics for hierarchical clustering:

**Dasgupta Cost** [Dasgupta 2016]

\[
\text{Das}(\hat{\mathcal{T}}) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum_{z} \mathbb{I}_{z=v_i \land v_j} c(z)
\]

**Tree-Sampling Divergence (TSD)** [Charpentier and Bonald 2019]

\[
\text{TSD}(\hat{\mathcal{T}}) = \text{KL}(p(z) || q(z)), \text{ where}
\]

\[
p(z) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \mathbb{I}_{z=v_i \land v_j}
\]

\[
q(z) = \sum_{v_i, v_j \in V} P(v_i, )P(v_j) \mathbb{I}_{z=v_i \land v_j}
\]
Hierarchical Clustering Metrics

Two established internal metrics for hierarchical clustering:

**Dasgupta Cost** [Dasgupta 2016]

\[
\text{Das}(\hat{f}) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum_z \mathbb{I}[z=v_i \wedge v_j] c(z)
\]

**Tree-Sampling Divergence (TSD)** [Charpentier and Bonald 2019]

\[
\text{TSD}(\hat{f}) = \text{KL}(p(z)||q(z)), \text{where}
\]

\[
p(z) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \mathbb{I}[z=v_i \wedge v_j]
\]

\[
q(z) = \sum_{v_i, v_j \in \mathcal{V}} P(v_i, v_j) P(v_j) \mathbb{I}[z=v_i \wedge v_j]
\]

Lowest common ancestors (LCAs) occur in both metrics.
Hierarchical Clustering Metrics: Relaxation

**Relaxed Dasgupata Cost**

\[
\text{Soft-Das}(\hat{f}) = \sum_{v_i,v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum_z P(z = v_i \land v_j) c(z)
\]

**Relaxed Tree-Sampling Divergence (TSD)**

\[
\text{Soft-TSD}(\hat{f}) = \text{KL}(p(z) || q(z)), \text{ where }
\]

\[
p(z) = \sum_{v_i,v_j \in \mathcal{E}} P(v_i, v_j) P(z = v_i \land v_j)
\]

\[
q(z) = \sum_{v_i,v_j \in V} P(v_i) P(v_j) P(z = v_i \land v_j)
\]

Indicators replaced by expectations (probabilities) \(\mathbb{I}_{[z=v_i \land v_j]}\) to \(P(z = v_i \land v_j)\)
Efficient Computation of LCA Probabilities

- Computation of LCA probabilities given matrices $A$ and $B$ is nontrivial.

- Key result: we draw connections to absorbing Markov chains to derive efficient closed-form matrix equations to compute LCA probabilities:
Efficient Computation of LCA Probabilities

- Computation of **LCA probabilities** given matrices $A$ and $B$ is nontrivial.

- **Key result**: we draw connections to **absorbing Markov chains** to derive **efficient closed-form matrix equations** to compute LCA probabilities:

\[ P_{v_i \neq v_j}^{LCA} \in \mathbb{R}^n = (P_{v_i}^{\text{anc}} \odot P_{v_j}^{\text{anc}})^T \cdot (I + \tilde{P}^{\text{anc}} \odot \tilde{P}^{\text{anc}})^{-1} \]

where $\odot$ denotes the element-wise (Hadamard) product. (See proof in App. A.9)

**Complexity** for the whole graph: $O(\#\text{internal nodes}^2 \cdot \#\text{edges})$
Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain</td>
<td>618.81</td>
<td>777.14</td>
<td>712.33</td>
<td>571.64</td>
<td>749.40</td>
<td>556.57</td>
<td>556.64</td>
<td><strong>503.67</strong></td>
<td>31.72</td>
<td>29.28</td>
<td>28.61</td>
<td>17.48</td>
<td>24.18</td>
<td>22.05</td>
<td>28.88</td>
</tr>
<tr>
<td>OpenFlight</td>
<td>382.45</td>
<td>633.66</td>
<td>393.58</td>
<td>463.43</td>
<td>487.96</td>
<td>488.90</td>
<td>363.40</td>
<td><strong>355.61</strong></td>
<td>55.48</td>
<td>51.51</td>
<td>53.89</td>
<td>59.08</td>
<td>49.50</td>
<td>39.56</td>
<td>52.05</td>
</tr>
<tr>
<td>Genes</td>
<td>202.17</td>
<td>247.26</td>
<td>251.01</td>
<td>495.26</td>
<td>366.53</td>
<td>247.07</td>
<td><strong>196.51</strong></td>
<td>77.16</td>
<td>69.43</td>
<td>68.45</td>
<td>67.40</td>
<td>37.22</td>
<td>57.61</td>
<td>50.61</td>
<td>67.99</td>
</tr>
<tr>
<td>Citeseer</td>
<td>92.27</td>
<td>178.23</td>
<td>98.61</td>
<td>215.62</td>
<td>150.26</td>
<td>131.89</td>
<td>84.04</td>
<td><strong>254.78</strong></td>
<td>56.47</td>
<td>57.51</td>
<td>53.06</td>
<td>30.73</td>
<td>46.76</td>
<td>42.68</td>
<td>55.41</td>
</tr>
<tr>
<td>Cora-ML</td>
<td>281.82</td>
<td>356.86</td>
<td>342.86</td>
<td>442.09</td>
<td>411.49</td>
<td>350.00</td>
<td>297.03</td>
<td><strong>262.48</strong></td>
<td>27.54</td>
<td>25.93</td>
<td>25.23</td>
<td>22.21</td>
<td>23.94</td>
<td>19.41</td>
<td>25.29</td>
</tr>
<tr>
<td>PolBlogs</td>
<td>377.63</td>
<td>443.48</td>
<td>350.74</td>
<td>320.58</td>
<td>354.86</td>
<td>433.77</td>
<td>364.14</td>
<td><strong>537.95</strong></td>
<td>45.28</td>
<td>46.03</td>
<td>43.40</td>
<td>32.02</td>
<td>39.70</td>
<td>38.39</td>
<td>43.23</td>
</tr>
<tr>
<td>WikiPhysics</td>
<td>736.11</td>
<td>986.32</td>
<td>753.81</td>
<td>759.07</td>
<td>840.15</td>
<td>740.87</td>
<td>658.04</td>
<td><strong>14,354</strong></td>
<td>36.77</td>
<td>37.75</td>
<td>24.75</td>
<td>26.05</td>
<td>25.21</td>
<td>33.70</td>
<td><strong>39.66</strong></td>
</tr>
<tr>
<td>ogbn-arxiv</td>
<td>22,870</td>
<td>31,655</td>
<td>52,666</td>
<td>OOM</td>
<td>22,076</td>
<td>24,077</td>
<td>20,760</td>
<td><strong>13,493</strong></td>
<td>45.33</td>
<td>46.12</td>
<td>27.90</td>
<td>24.80</td>
<td>34.07</td>
<td>45.40</td>
<td><strong>48.36</strong></td>
</tr>
<tr>
<td>ogbl-collab</td>
<td>13,835</td>
<td>20,664</td>
<td>91,807</td>
<td>OOM</td>
<td>34,934</td>
<td>21,057</td>
<td>15,714</td>
<td><strong>31,138</strong></td>
<td>40.74</td>
<td>44.43</td>
<td>20.21</td>
<td>OOM</td>
<td>15.96</td>
<td>27.82</td>
<td>38.97</td>
</tr>
</tbody>
</table>

Table 1: Hierarchical clustering results ($n' = 512$). Bold/underline indicate best/second best scores.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FPH</th>
<th>DCSBM</th>
<th>DW</th>
<th>Ad./Ad.</th>
<th>VGAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora-ML</td>
<td>95.7</td>
<td>95.5</td>
<td>94.3</td>
<td>86.5</td>
<td><strong>95.9</strong></td>
</tr>
<tr>
<td>Citeseer</td>
<td><strong>96.2</strong></td>
<td>93.6</td>
<td>96.0</td>
<td>76.8</td>
<td>94.8</td>
</tr>
<tr>
<td>PolBlogs</td>
<td>94.3</td>
<td><strong>94.9</strong></td>
<td>84.8</td>
<td>92.6</td>
<td>92.8</td>
</tr>
<tr>
<td>WikiPhysics</td>
<td><strong>97.2</strong></td>
<td>96.9</td>
<td>92.9</td>
<td>96.6</td>
<td>97.0</td>
</tr>
<tr>
<td>Brain</td>
<td>94.1</td>
<td><strong>95.2</strong></td>
<td>83.8</td>
<td>90.7</td>
<td>93.2</td>
</tr>
<tr>
<td>OpenFlight</td>
<td><strong>99.3</strong></td>
<td>99.0</td>
<td>94.3</td>
<td>98.4</td>
<td>99.0</td>
</tr>
<tr>
<td>Genes</td>
<td>69.8</td>
<td>66.9</td>
<td><strong>70.3</strong></td>
<td>53.0</td>
<td>66.6</td>
</tr>
</tbody>
</table>

Table 4: AUC-PR score (%) for link prediction.