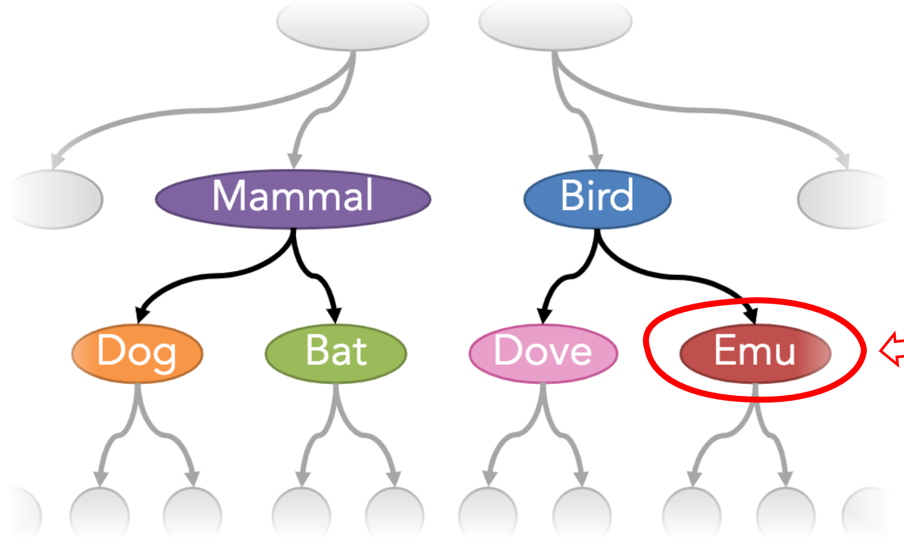
UMassAmherst | College of Information & Computer Sciences

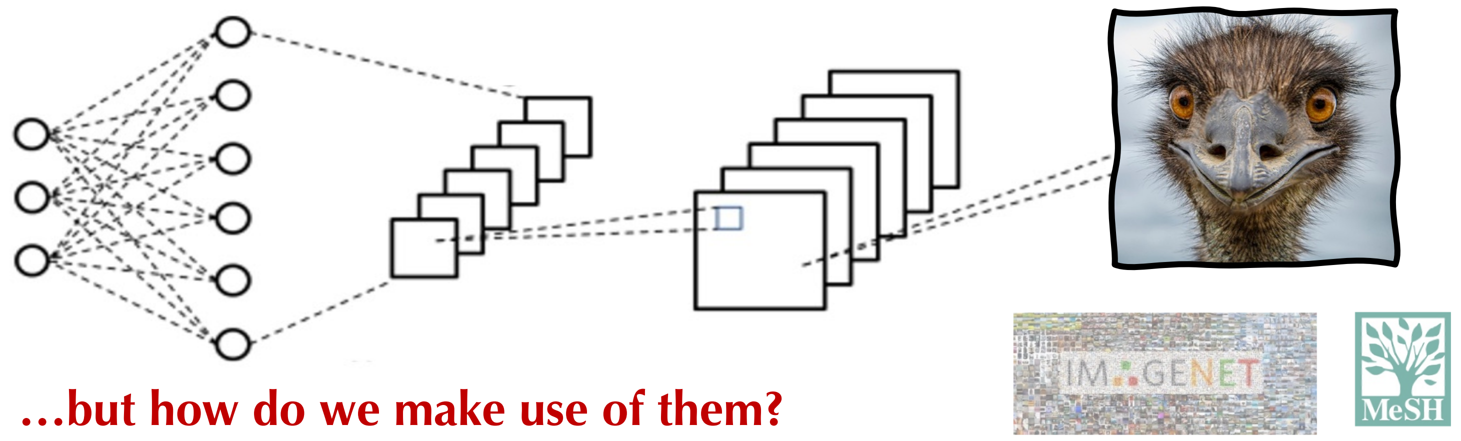
Learning Representations for Hierarchies with Minimal Support

Google DeepMind

Benjamin Rozonoyer, Michael Boratko, Dhruvesh Patel, Wenlong Zhao, Shib Sankar Dasgupta, Hung Le & Andrew McCallum



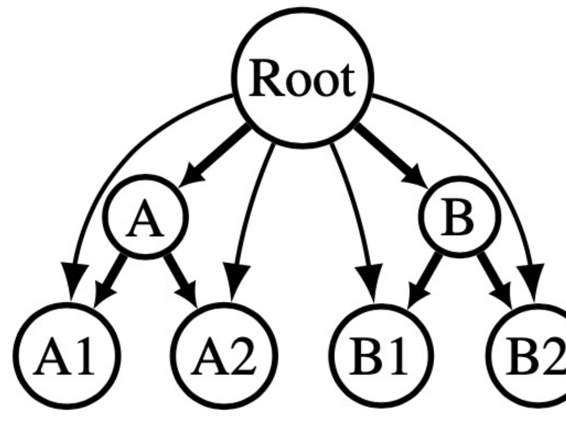
Taxonomies are often available for machine learning data…



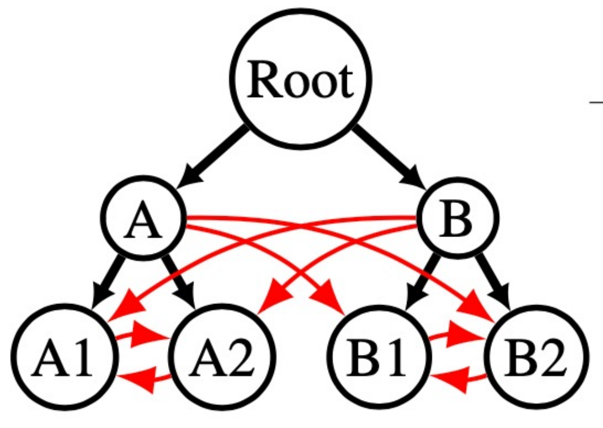
…but how do we make use of them?

Randomly sampling label pairs is inefficient. Can we leverage the structure?

Question: *What is the* ***smallest subset of entries*** *of the adjacency matrix* ***necessary to uniquely determine a graph****?*



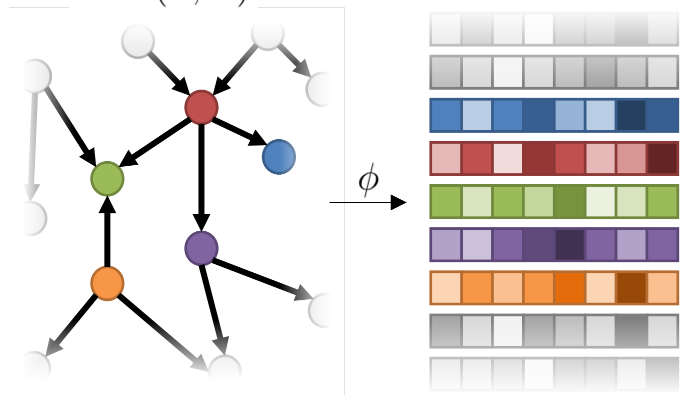
	Root	A	B	A1	A2	B1	B2
Root	0	**1**	**1**	1	1	1	1
A	0	0	0	**1**	**1**	0	0
B	0	0	0	0	0	**1**	**1**
A1	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0
B1	0	0	0	0	0	0	0
B2	0	0	0	0	0	0	0



	Root	A	B	A1	A2	B1	B2
Root		1	1				
A				1	1	0	0
B				0	0	1	1
A1					0		
A2				0			
B1							0
B2						0	

Answer: ***If we know the graph has a particular property*** $\mathcal{P}$ *(e.g. symmetry,* ***hierarchy****),* ***we don't need all entries***

Node Embeddings for Representing Graph Structure



$\mathrm{E}_\phi(\bullet, \bullet) = f(\phi(\bullet), \phi(\bullet))$

$P(\bullet \to \bullet) = \exp\left[-\mathrm{E}_\phi(\bullet, \bullet)\right]$

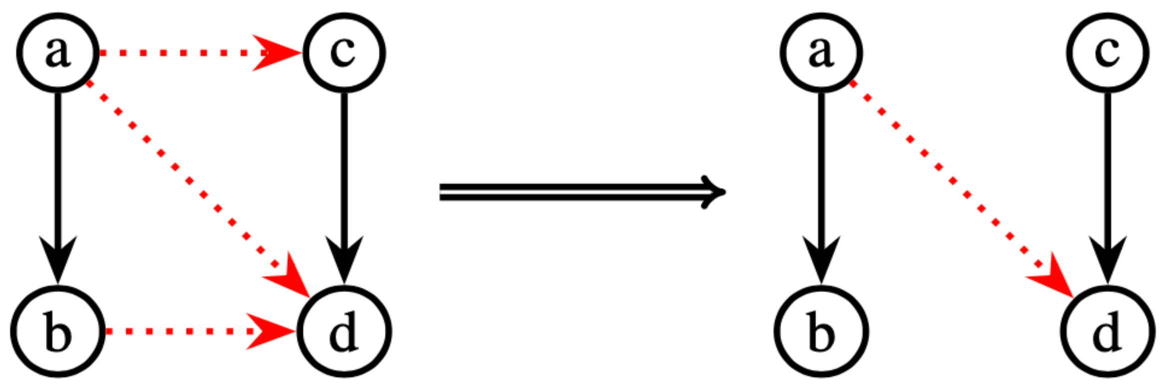
$\exists \tau : \bullet \to \bullet \in E \iff \mathrm{E}_\phi(\bullet, \bullet) < \tau$

Naïve Loss

$$\mathcal{L}_{\text{full}}(\theta; G) := \sum_{(u\to v)\in E} \ell^+(\mathrm{E}_\theta(u,v)) + \sum_{(u\to v)\in \overline{E}} \ell^-(\mathrm{E}_\theta(u,v))$$

Distinguishing Digraphs via Signed Digraphs

1. Transitive reduction of positive edges is sufficient
2.



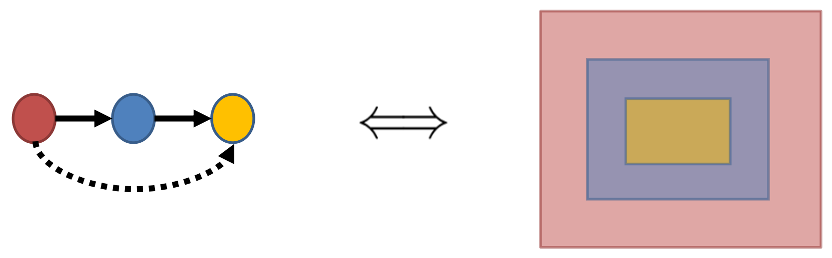
Algorithm 1 FINDMINDISTINGUISHER

Require: $G = (V, E)$ transitively-closed DAG

1: $E^* \leftarrow \overline{E}$
2: **for** $(a \to d) \in \overline{E}$ **do**
3: **for** $(a \to b) \in E$ **do**
4: $E^* \leftarrow E^* \setminus \{(b \to d)\}$
5: **end for**
6: **for** $(c \to d) \in E$ **do**
7: $E^* \leftarrow E^* \setminus \{(a \to c)\}$
8: **end for**
9: **end for**
10: **return** $H^* = (V, E^{\text{tr}}, E^*)$

Transitivity Bias

Box embeddings have transitivity bias!



$\iff P(\bullet|\bullet) = 1 \;\wedge\; P(\bullet|\bullet) = 1 \;\wedge\; P(\bullet|\bullet) = 1$

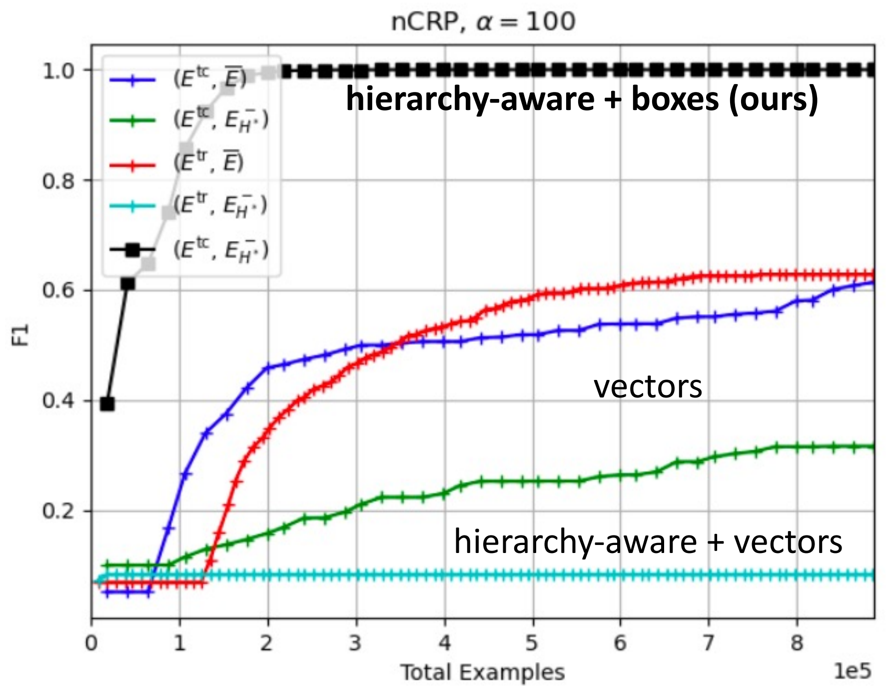
Transitivity bias allows us to train on minimal distinguishing set of edges

Hierarchy-Aware Loss

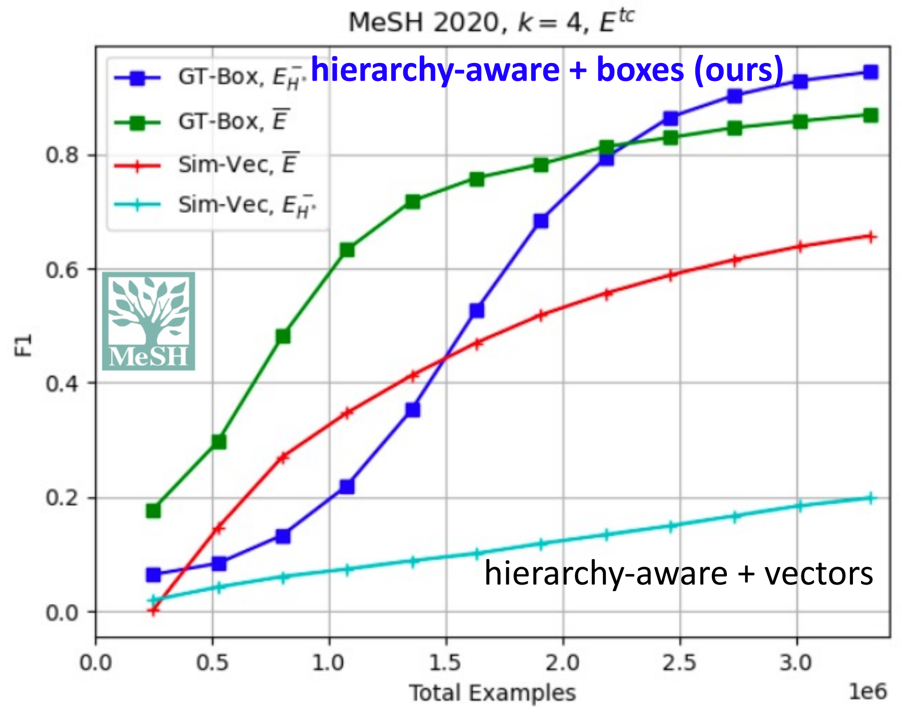
$$\mathcal{L}_{\text{ha}}(\theta; H) := \sum_{(u\to v)\in E_H^+} \ell^+(\mathrm{E}_\theta(u,v)) + \sum_{(u\to v)\in E_H^-} \ell^-(\mathrm{E}_\theta(u,v))$$

Experiments: F1 between Hierarchy and Model's Predictions

- Boxes (transitivity bias) vs Vectors (no transitivity bias)
- Sample positive edges from transitive reduction vs transitive closure
- Sample negative edges from minimal hierarchy-aware set vs edge complement



Box embeddings can take advantage of hierarchy-aware sampling



Hierarchy-aware sampling converges faster on large taxonomies

Conclusion

- Observing **as few as 1% of the entries** of the adjacency matrix uniquely identifies a hierarchy
- **Box embeddings…**
  - have **transitivity bias**, allowing them to take advantage of the **hierarchy-aware** set
  - **converge faster** with **smaller**, structurally-informed batches, to **higher accuracy** than vectors

[1] Boratko et al. "Capacity and bias of learned geometric embeddings for directed graphs." NeurIPS 34 (2021)
[2] Patel et al. "Modeling label space interactions in multi-label classification using box embeddings" ICLR (2022)