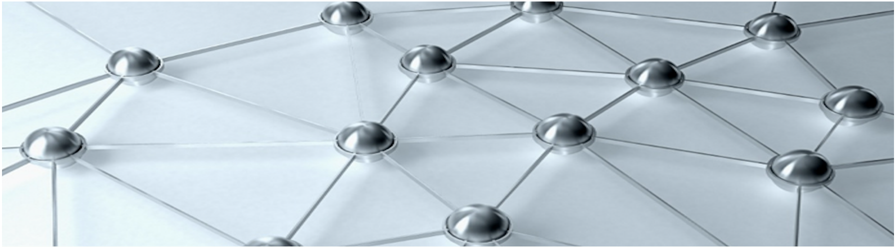


Node ranking in labeled networks

Chamalee Wickrama Arachchi, Nikolaj Tatti

University of Helsinki, Finland

SDM23

What do we do?

Explainable Node Ranking Problem

- Rank nodes according to some hierarchy.
- Giving some explanation for each hierarchy (rank).

Explainable node ranking problem

Input:

- Directed graph. (Weighted or unweighted)
- Node labels.
- Input parameter: k.

Output:

- Rank the nodes.
- Assign an integer to each node between 0 and $k - 1$.

Such that:

- A penalty score is minimized.
- Each rank is explained by a set of labels.

Agony score ($q(r, G)$)

- r(i): The rank of node i.
- Edge (u, v) is forward: if $r(u) < r(v)$.
- Edge (u, v) is backward: if $r(u) \geq r(v)$.

- Agony score: penalizes the backward edges (u, v) based on the difference between the ranks $r(u) - r(v)$.
  - penalty function $p(d) = \max(0, d + 1)$ where d is the difference between ranks.
  - Example:
  - $r(u) = r(v)$ gives $q(r) = 1$.
  - $r(u) = r(v) + 1$ gives $q(r) = 2$.

Node Ranking Problem (No labels)

- Given a directed graph $G = (V, E)$ and an integer k, find a rank assignment r minimizing the agony such that $0 \leq r(v) \leq k - 1$ for every $v \in V$.

- This problem is polynomial time solvable.
  - Unweighted problem.[1]
  - Weighted problem.[2]

---

[1] Gupte, M., Shankar, P., Li, J., Muthukrishnan, S. and Iftode, L., 2011, March. Finding hierarchy in directed online social networks. In Proceedings of the 20th international conference on World wide web (pp. 557-566).

[2] Tatti, N., 2017. Tiers for peers: a practical algorithm for discovering hierarchy in weighted networks. Data mining and knowledge discovery, 31, pp.702-738.

Node ranking with labels

Our Approach:

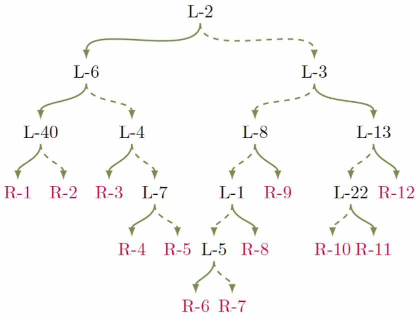
1. Construct a label tree.
2. Find explainable hierarchies from using label tree.

Complexity:

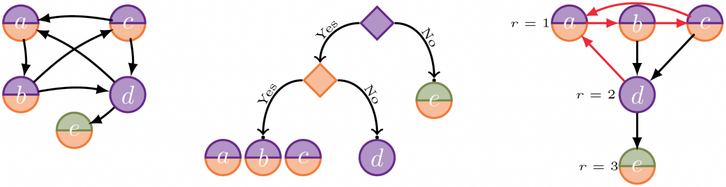
- It turns out that finding agony minimizing label tree is **NP**-Hard.

Label tree

- Binary tree like structure.
- Each leaf represents a rank.
- Each non-leaf represents a label.
- Each leaf or non-leaf contains a Boolean value. (criterion)
- Ordered tree: Left leaves with lower ranks, Right leaves with higher ranks.



Toy example



(a) Input (b) Label tree (c) Explainable ranks

Figure: Explainable ranking

Greedy Algorithm - Main loop

Finding the label tree which minimizes the agony is computationally intractable.

- Greedy heuristic.
- Divide-and-conquer approach.

```
find optimal label t and criterion c
  using TEST;
Δ ← reduction in score when
  splitting with (t, c);
if Δ < 0 then
    β, γ ← CONSTRUCT(α, t, c);
    GREEDY(β); GREEDY(γ);
```

Subroutines:

- TEST: Gain due to split of leaf α, with label t.
- CONSTRUCT: Updating counters.

Cardinality constraint

Enforce number of ranks to be k:

- Prune the tree.
- Dynamic programming approach.

DP recurrence formula:

- β: The left child.
- γ: The right child.
- $o(\alpha; h)$: The optimal gain that obtained in the branch in T starting from the node α by using only h leaves.
- $\Delta(\alpha)$: The improvement in agony as recorded in GREEDY algorithm.

$$o(\alpha; h) = \Delta(\alpha) + \min_{1 \leq \ell \leq h-1} o(\beta; \ell) + o(\gamma; h - \ell)$$

Experiments with Synthetic Datasets

- θ - The percentage of vertices which contains false labels.
- μ - The percentage of vertices which contains noise labels.
- η - The percentage of forward edges.
- h - The number of ranks.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	h	θ	μ	η
Syn-1	4 000	3 200	5	0.04	0.05	0.6
Syn-2	5 000	7 000	8	0.04	0.06	0.7
Syn-3	1 200	1 200	5	0.05	0.07	0.65
Syn-4	1 000	1 750	6	0.05	0.08	0.9
Syn-5	4 000	4 200	7	0.06	0.09	0.85
Syn-6	1 000	5 600	15	0.08	0.1	0.8
Syn-7	40 000	135 000	10	0.08	0.11	0.75

Experiments with Synthetic Datasets

- q_{true} and q_{dis} : The ground truth and discovered agony scores
- q_{base} is the discovered agony excluding labels.
- h_{dis} and h_{base} : The discovered number of ranks using our algorithm and baseline.
- k_{tau} is the Kendall's τ coefficient.
- $time$ gives the computational time in seconds for our algorithm.

Dataset	q_{true}	q_{dis}	q_{base}	h_{dis}	h_{base}	$k\tau_{dis}$	$k\tau_{base}$	$time$
Syn-1	2 242	2 322	10	5	25	0.97	0.1	2.78
Syn-2	4 160	4 453	648	8	47	0.97	0.29	6.47
Syn-3	782	798	2	5	20	0.97	0.15	0.5
Syn-4	388	535	2	7	27	0.96	0.58	0.62
Syn-5	1 268	1 591	16	8	22	0.96	0.26	4.21
Syn-6	2 262	3 237	2 158	15	19	0.95	0.96	1.29
Syn-7	66 978	89 017	51 504	10	24	0.93	0.79	189.95

Effect of noise

- θ - The percentage of vertices which contains false labels.
- μ - The percentage of vertices which contains noise labels.
- η - The percentage of forward edges.

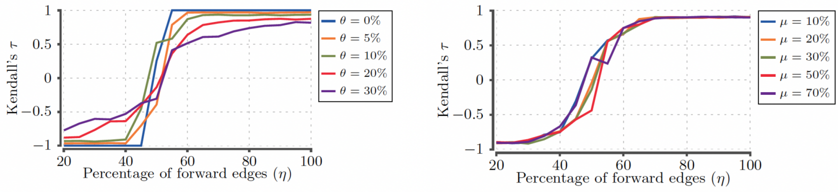
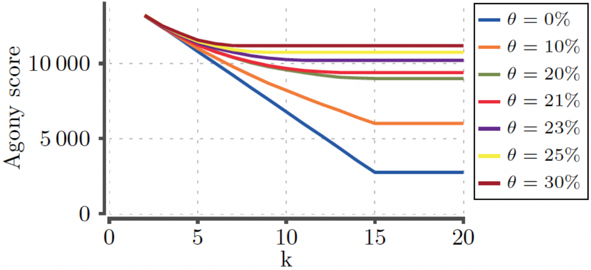


Figure: Kendall's τ coefficient as a function of η for the cases of several false label probabilities (θ) (left) and noise label probabilities (μ) (right). Experimental setting: $|V| = 4\,000$, $|E| = 7\,000$, $\mu = 0.05$, and $h = 10$.

Effect of cardinality constraint

- θ - The percentage of vertices which contains false labels.
- k - Number of ranks.



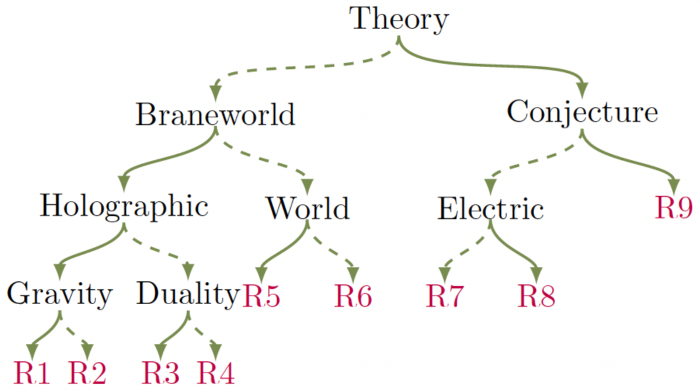
Experiments with Real-world Datasets

- $|T|$: The number of labels.
- d : The height of the label tree.

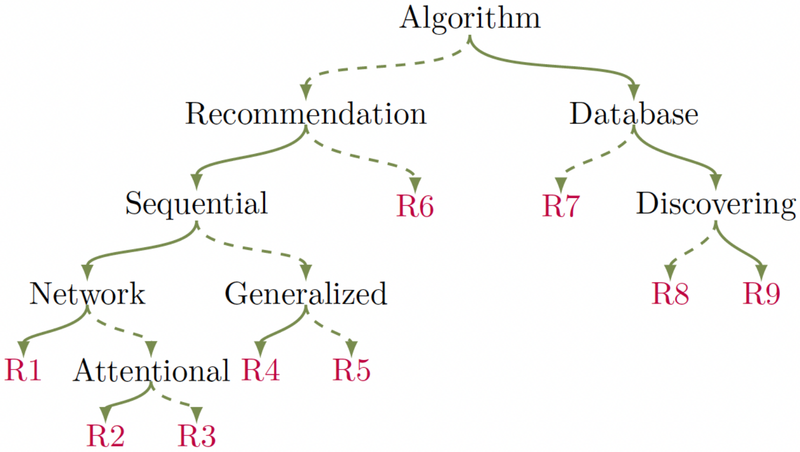
Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$
EIES	32	460	9
School	329	502	9
Cora	2 708	5 429	7
Cite-seer	3 264	4 536	6
Patent-citation	5 439	4 232	394
DBLP-citation	30 581	70 972	10 245
Physics-citation	29 555	352 807	4 715

Dataset	q_{dis}	q_{base}	h_{dis}	h_{base}	d	*time*
EIES	14 035	12 682	4	3	4	3.4ms
School	1 426	982	3	10	3	20ms
Cora	5 351	340	3	18	3	35s
Cite-seer	4 397	0	3	16	3	1.3s
Patent-citation	4 074	0	7	3	4	4m
DBLP-citation	68 511	329	10	38	6	43m
Physics-citation	339 475	2 268	9	236	5	22m

Case study : Physics citation dataset



Case study : DBLP dataset



Conclusion

- Introduced a novel tree-based, hierarchy mining problem for vertex-labeled, directed graphs.
- Our goal was to rank the nodes into tiers so that ideally the edges are directing from the lower ranks.
- The construction of such a label tree optimally was an **NP**-hard problem.
- We showed that this problem is inapproximable when we limit the number of leaves.
- Proposed a heuristic algorithm.
- Complexity of our algorithm is $\mathcal{O}((n+m)\log n + \ell R)$, where $R = \sum_v |L(v)|$ is the number of node-label pairs in given graph, ℓ is the number of nodes in the resulting label tree, and n and m are the number of nodes and edges respectively.
- The synthetic experiments showed that our approach accurately recovers the latent hierarchy.
- Showed experimentally with real-world datasets that label-driven algorithm achieved a good quality ranks and computational time is reasonable.

Thank you for your attention!!!