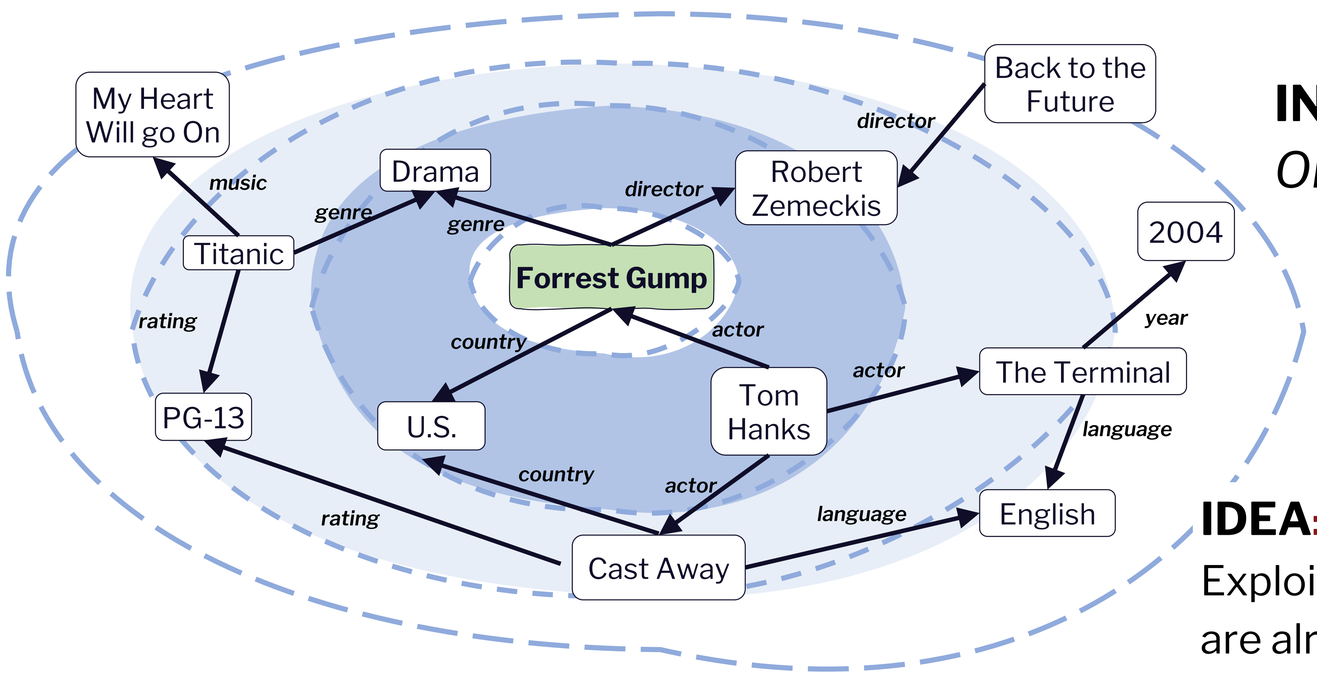
Personalized Page Rank on Knowledge Graphs: **Particle Filtering is all you need!**

Denis Gallo, Matteo Lissandrini, Yannis Velegrakis

UNITRENTO · Aalborg Universitet · Universiteit Utrecht

Fast and Scalable Top-k Personalized Page Rank Queries on Knowledge Graphs: *Measures a "proximity" within nodes in the graph*

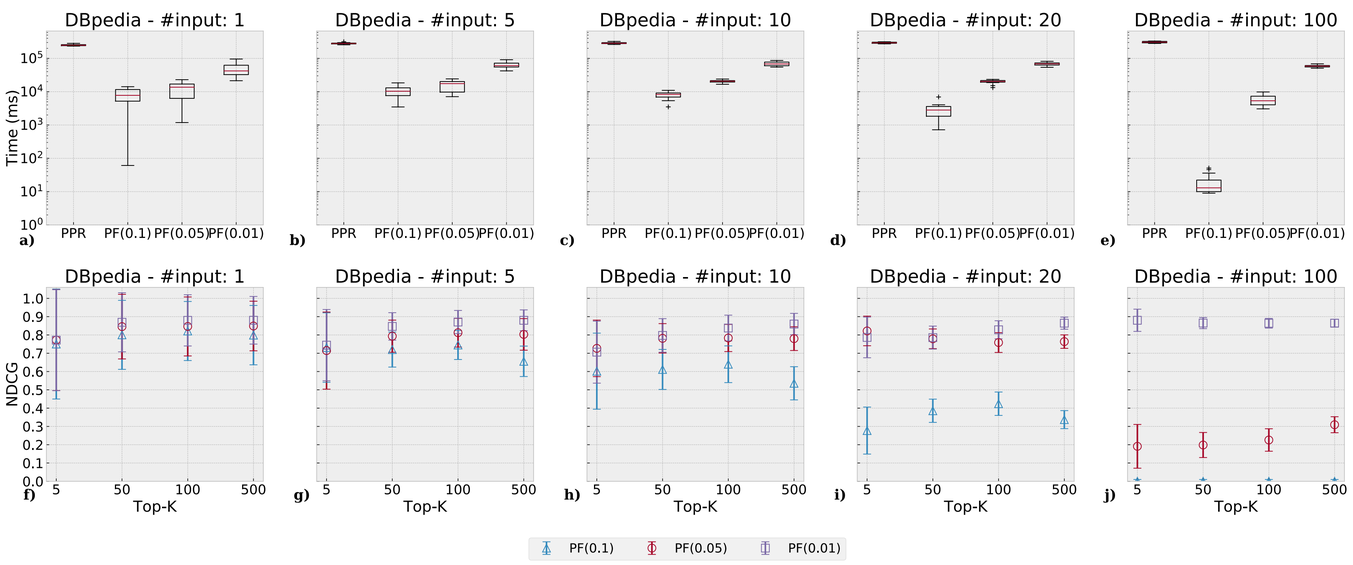


INPUT:
One or more nodes

OUTPUT:
Top-K Relevant nodes

IDEA:
Exploit what graph databases are already optimized for: ***navigating local connection***.

Compared over real world GDBMS: Neo4j



Particle Filtering consistently outperforms classical implementation by 1 to 4 orders of magnitude.

Particle Filtering

Simulate a set of particles navigating the graphs

Require: Graph G; Query nodes $\mathbf{Q}$
Require: Restart probability $c \in [0,1]$; Threshold $\tau \in [0,1]$
Require: Query value k
Ensure: Ranked Top-K nodes

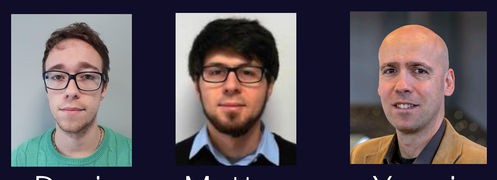
```
p ← {}
for each q_i ∈ Q do
    p[q_i] ← 1/τ                          ▷ Initialize Particles
while ∃ n_i ∈ p | p[n_i] ≠ 0 do
    temp ← {}
    for each n_i ∈ p | p[n_i] ≠ 0 do
        particles ← p[n_i] × (1 − c)
        for each e : (n_i → n_j) ∈ G do   ▷ Sorted by Weight
            if particles ≤ τ then
                break
            passing ← MAX(particles × e.weight(), τ)
            temp[n_j] ← temp[n_j] + passing
            particles ← particles − passing
    p ← temp
    for each n_i ∈ p do
        v[n_i] ← v[n_i] + p[n_i] × c      ▷ Update score
return top-k(v)
```

Particularly suitable for Knowledge Graphs.

Desirable features:

- multi-source queries
- heterogeneous edge weights
- top-k retrieval

Questions?
Ask the authors


Denis Matteo Yannis

We propose an optimized approximate algorithm to compute **Top-K Personalized PageRank** ranking of graph nodes. The algorithm uses the **"particle filtering"** technique, a combination of random walks and sampling, and is tailored to be included in existing GDBMS systems especially for **Knowledge Graph** exploration.

Full Paper & Code
Scan the QR code

http://j.mp/EDBT20PPR