



Graphs in Machine Learning

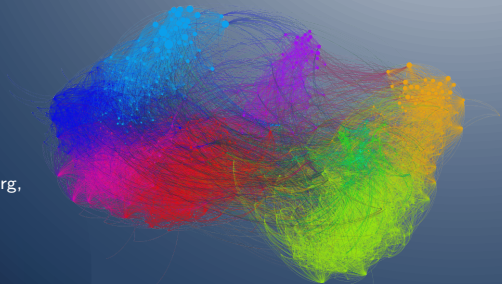
Movie Recommendations

Graph Distance Approaches

Michal Valko

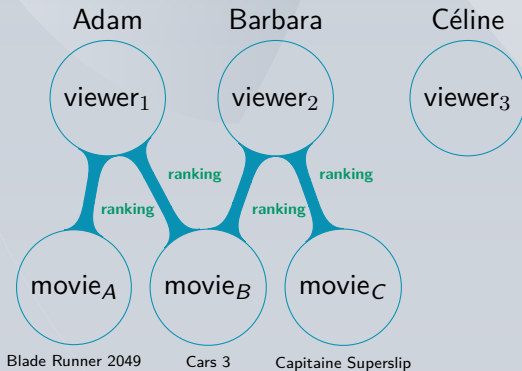
Inria & ENS Paris-Saclay, MVA

Partially based on material by: Ulrike von Luxburg,
Gary Miller, Doyle & Schnell, Daniel Spielman



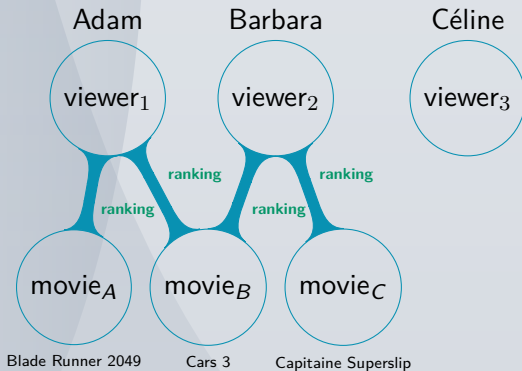
Use of Laplacians: Movie recommendation

How to do movie recommendation on a bipartite graph?



Use of Laplacians: Movie recommendation

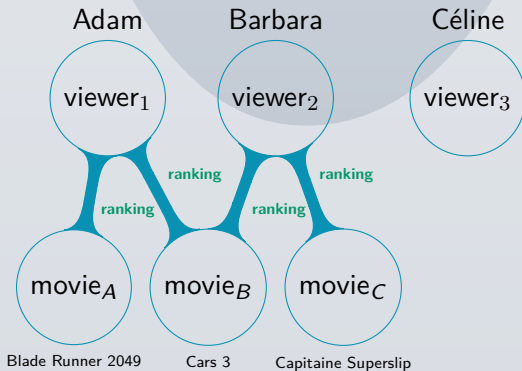
How to do movie recommendation on a bipartite graph?



Question: *Do we recommend Capitaine Superslip to Adam?*

Use of Laplacians: Movie recommendation

How to do movie recommendation on a bipartite graph?



Question: *Do we recommend Capitaine Superslip to Adam?*

Let's compute some $\text{score}(v, m)$!

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$?

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$? Using some **graph distance**!

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Problem: If there is a weak edge, the path should not be good.

Use of Laplacians: Movie recommendation

How to compute the score(v, m)? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Problem: If there is a weak edge, the path should not be good.

Idea₂: change the path weight

$$\text{score}_2(v, m) = \max_{vPm} \text{weight}_2(P) = \\ \max_{vPm} \min_{e \in P} \text{ranking}(e)$$

Use of Laplacians: Movie recommendation

How to compute the score(v, m)? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Problem: If there is a weak edge, the path should not be good.

Idea₂: change the path weight

$$\text{score}_2(v, m) = \max_{vPm} \text{weight}_2(P) = \\ \max_{vPm} \min_{e \in P} \text{ranking}(e)$$

Problem of 1&2: Additional paths does not improve the score.

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Problem: If there is a weak edge, the path should not be good.

Idea₂: change the path weight

$$\text{score}_2(v, m) = \max_{vPm} \text{weight}_2(P) = \max_{vPm} \min_{e \in P} \text{ranking}(e)$$

Problem of 1&2: Additional paths does not improve the score.

Idea₃: consider everything

$$\text{score}_3(v, m) = \text{max flow from } m \text{ to } v$$

Use of Laplacians: Movie recommendation

How to compute the $\text{score}(v, m)$? Using some **graph distance**!

Idea₁: maximally weighted path

$$\text{score}(v, m) = \max_{vPm} \text{weight}(P) = \max_{vPm} \sum_{e \in P} \text{ranking}(e)$$

Problem: If there is a weak edge, the path should not be good.

Idea₂: change the path weight

$$\text{score}_2(v, m) = \max_{vPm} \text{weight}_2(P) = \max_{vPm} \min_{e \in P} \text{ranking}(e)$$

Problem of 1&2: Additional paths does not improve the score.

Idea₃: consider everything

$$\text{score}_3(v, m) = \text{max flow from } m \text{ to } v$$

Michal Valko

`michal.valko@inria.fr`

Inria & ENS Paris-Saclay, MVA

`https://misovalko.github.io/mva-ml-graphs.html`