QuickRank

A Recursive Ranking Algorithm

Amy Greenwald and John R. Wicks

amy@cs.brown.edu and jwicks@cs.brown.edu

Brown University

Introduction

- We approach ranking of individuals in a society as problem in aggregation of preferences, akin to voting.
  - Except the individuals and the alternatives are the same.
  - That is, preferences are expressed as importance judgments made by each person of everyone else.

Introduction

- We approach ranking of individuals in a society as problem in aggregation of preferences, akin to voting.
  - Except the individuals and the alternatives are the same.
  - That is, preferences are expressed as importance judgments made by each person of everyone else.
- We assume that hierarchies exist:

Introduction

- We approach ranking of individuals in a society as problem in aggregation of preferences, akin to voting.
  - Except the individuals and the alternatives are the same.
  - That is, preferences are expressed as importance judgments made by each person of everyone else.
- We assume that hierarchies exist:

  > *Almost all societies have elementary units called families, which may be grouped into villages or tribes, and these into larger groupings, and so on. If we make a chart of social interactions, of who talks to whom, the clusters of dense interaction in the chart will identify a rather well-defined hierarchic[al] structure.* Simon [1962]

Introduction

- We believe they should be exploited, because:

Introduction

- We believe they should be exploited, because:
  - they may imply limitations of the given importance judgments (eg. subcommunities)

Introduction

- We believe they should be exploited, because:
  - they may imply limitations of the given importance judgments (eg. subcommunities)
  - hierarchies $\rightarrow$ recursion $\rightarrow$ parallelizeability $\rightarrow$ scaleability

Introduction

- We believe they should be exploited, because:
  - they may imply limitations of the given importance judgments (eg. subcommunities)
  - hierarchies $\rightarrow$ recursion $\rightarrow$ parallelizeability $\rightarrow$ scaleability
- QuickRank converts *any* ranking algorithm into one which respects a given hierarchy, based on:

Introduction

- We believe they should be exploited, because:
  - they may imply limitations of the given importance judgments (eg. subcommunities)
  - hierarchies $\rightarrow$ recursion $\rightarrow$ parallelizeability $\rightarrow$ scaleability
- QuickRank converts *any* ranking algorithm into one which respects a given hierarchy, based on:
  - the Peer-Review Principle

Introduction

- We believe they should be exploited, because:
  - they may imply limitations of the given importance judgments (eg. subcommunities)
  - hierarchies $\rightarrow$ recursion $\rightarrow$ parallelizeability $\rightarrow$ scaleability
- QuickRank converts *any* ranking algorithm into one which respects a given hierarchy, based on:
  - the Peer-Review Principle
  - Bonacich's Hypothesis

Review of PageRank

C

A

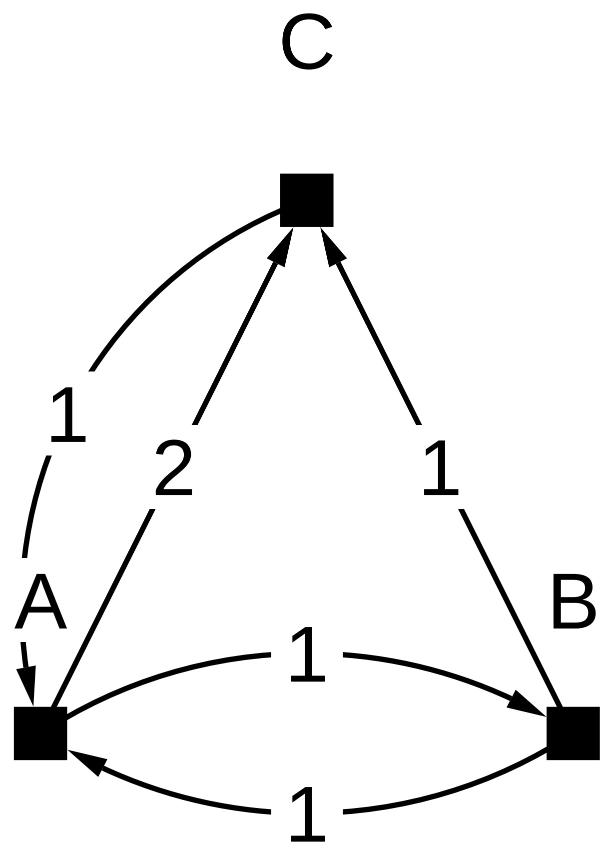
A

C, B, C

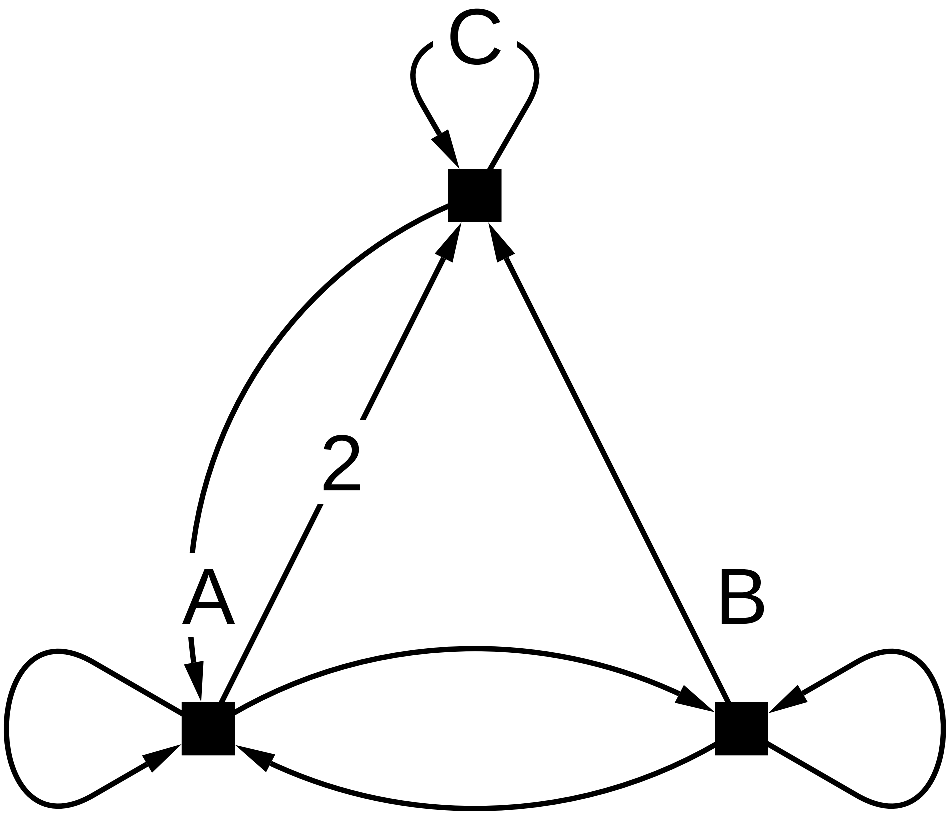
B

A, C

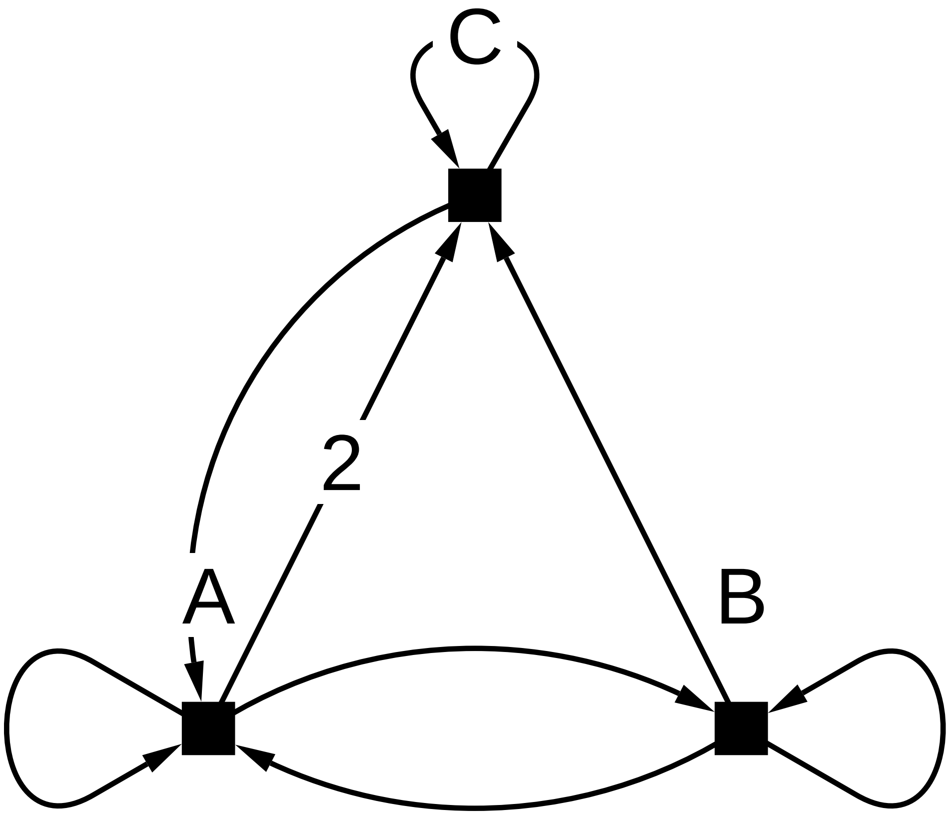
Review of PageRank



Review of PageRank

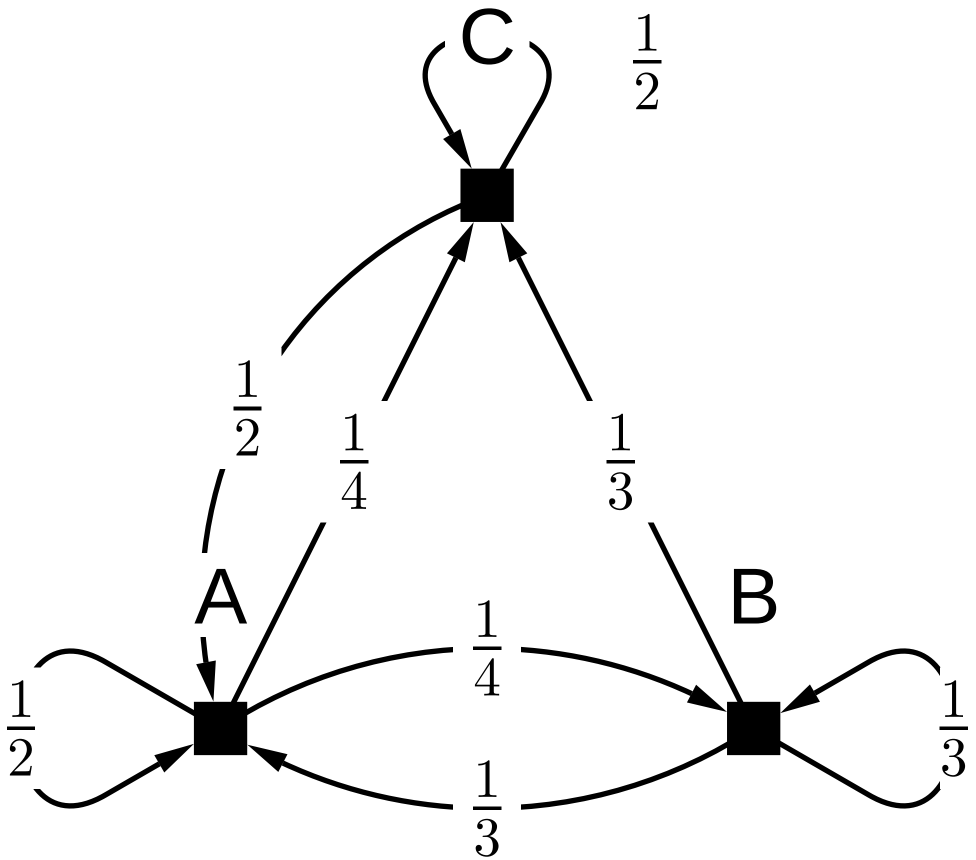


Review of PageRank



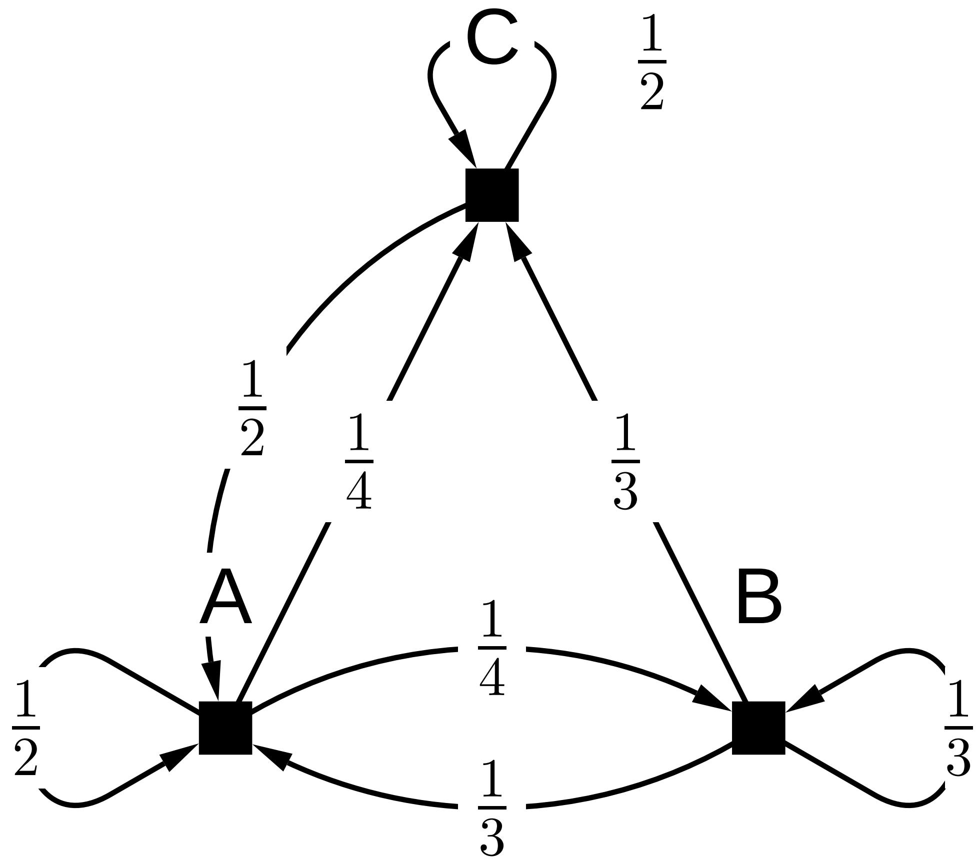
A	B	C
1	2	1

Review of PageRank



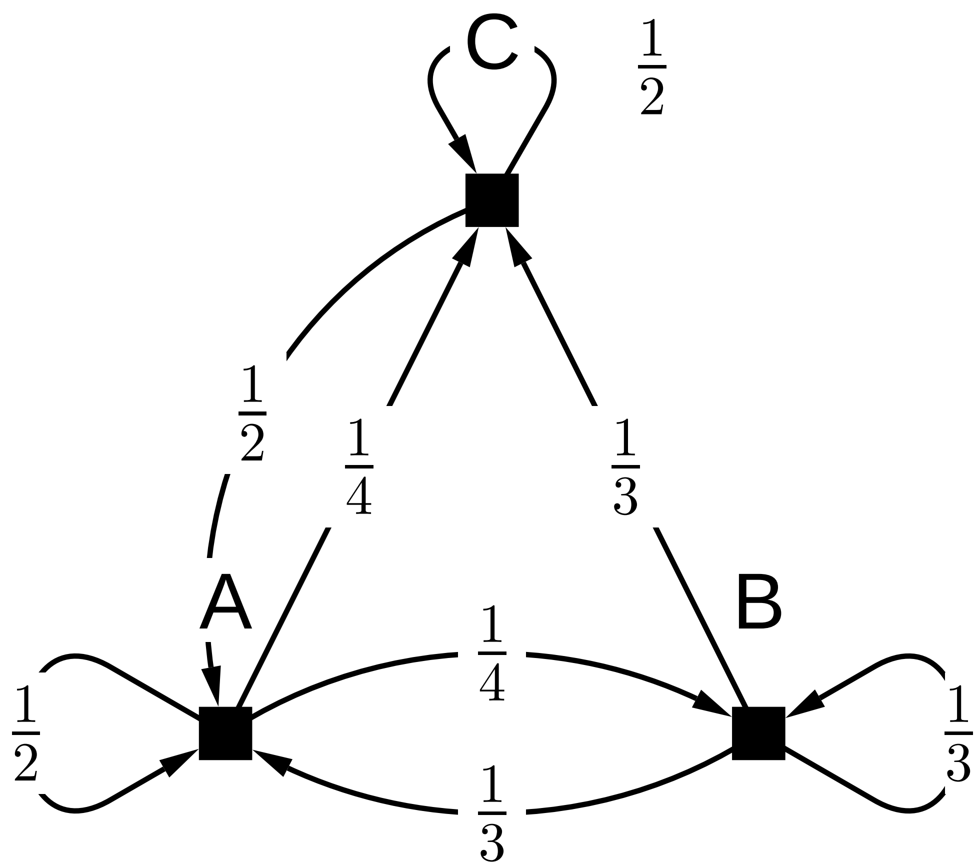
A	B	C
0.25	0.50	0.25

Review of PageRank



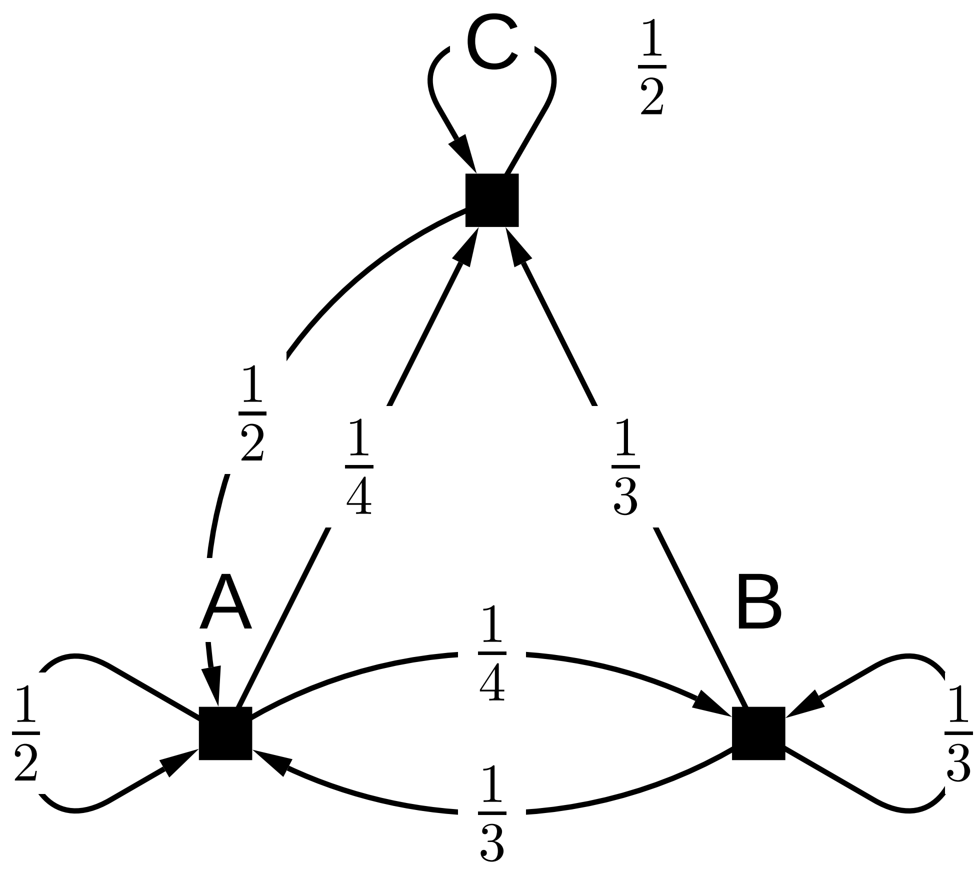
A	B	C
0.30	0.37	0.33

Review of PageRank



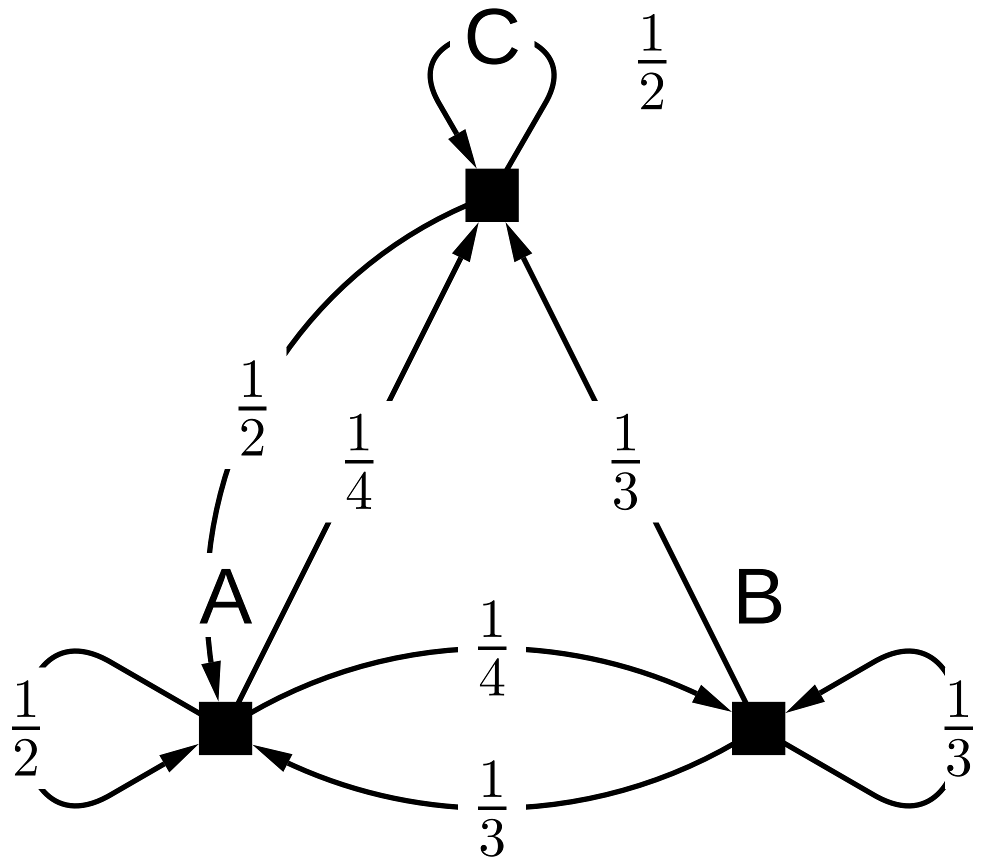
A	B	C
0.32	0.32	0.36

Review of PageRank



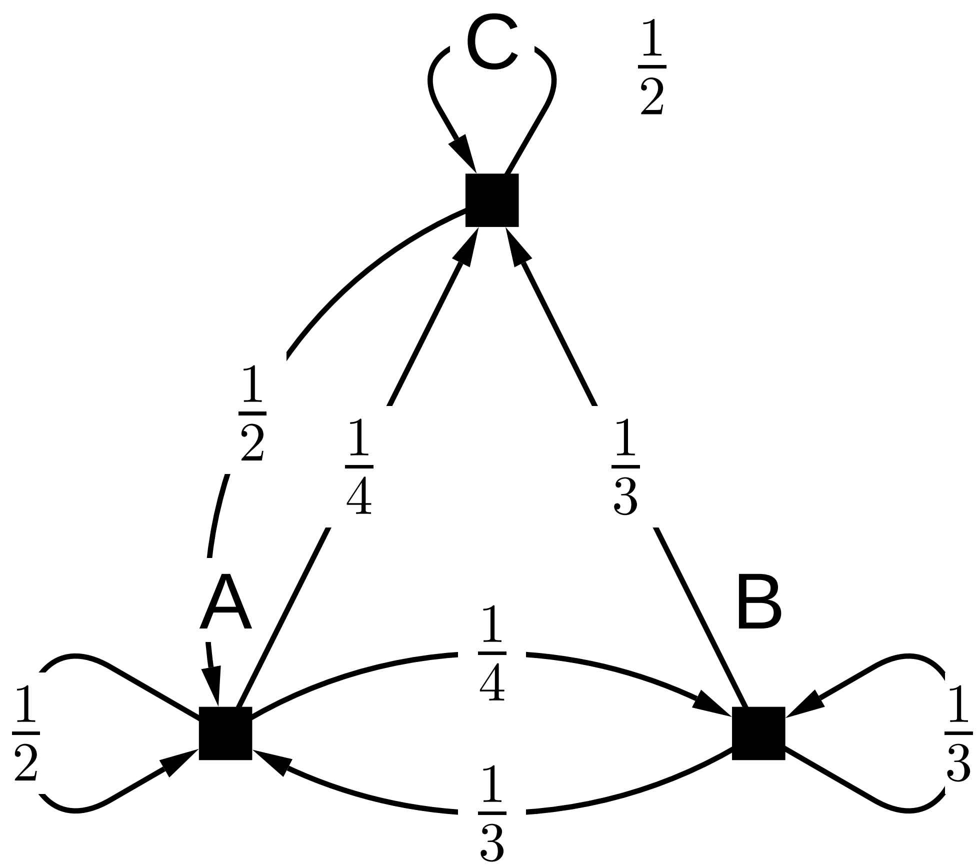
A	B	C
0.33	0.28	0.39

Review of PageRank



A	B	C
0.36	0.21	0.43

Review of PageRank



A	B	C
2	3	1

Key Points

- PageRank considers a ranking as specified by a probability distribution.

Key Points

- PageRank considers a ranking as specified by a probability distribution.
- It takes a collection of judgments and a prior ranking as input and produces a posterior ranking as output.

Key Points

- PageRank considers a ranking as specified by a probability distribution.
- It takes a collection of judgments and a prior ranking as input and produces a posterior ranking as output.
- Fixing an enumeration of individuals, a ranking algorithm corresponds to a mapping:

$$\begin{array}{ccccc} \text{Pos. Matrix} & \times & \text{Prob. Dist.} & \to & \text{Prob. Dist.} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} & & \begin{bmatrix} 0.25 \\ 0.50 \\ 0.25 \end{bmatrix} & & \begin{bmatrix} 0.36 \\ 0.21 \\ 0.43 \end{bmatrix} \end{array}$$

Key Points

- PageRank considers a ranking as specified by a probability distribution.
- It takes a collection of judgments and a prior ranking as input and produces a posterior ranking as output.
- Fixing an enumeration of individuals, a ranking algorithm corresponds to a mapping:

$$\underset{\text{Pos. Matrix}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}} \times \underset{\text{Prob. Dist.}}{\begin{bmatrix} 0.25 \\ 0.50 \\ 0.25 \end{bmatrix}} \rightarrow \underset{\text{Prob. Dist.}}{\begin{bmatrix} 0.36 \\ 0.21 \\ 0.43 \end{bmatrix}}$$

- We view this as aggregation of importance judgments to yield a collective ranking.

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

  This suggests that the collective (posterior) ranking should incorporate the judgment made by each individual, weighted by its (prior) rank.

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

  This suggests that the collective (posterior) ranking should incorporate the judgment made by each individual, weighted by its (prior) rank.

- With a uniform prior:

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

  This suggests that the collective (posterior) ranking should incorporate the judgment made by each individual, weighted by its (prior) rank.

- With a uniform prior:

$$\frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} =$$

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

  This suggests that the collective (posterior) ranking should incorporate the judgment made by each individual, weighted by its (prior) rank.

- With a uniform prior:

$$\frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}/3 = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}/3 \rightarrow \begin{bmatrix} \frac{3}{11} \\ \frac{2}{11} \\ \frac{4}{11} \end{bmatrix}$$

A Simple Ranking Algorithm

- **Bonacich's hypothesis**

  An individual is important if he is deemed important by other important individuals.

  This suggests that the collective (posterior) ranking should incorporate the judgment made by each individual, weighted by its (prior) rank.

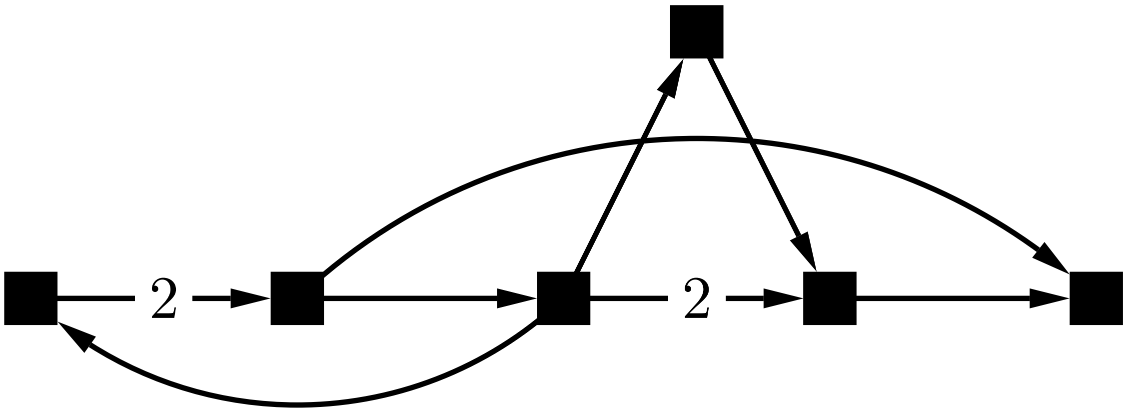
- With a uniform prior:

$$\frac{1}{3}\begin{bmatrix}1\\1\\2\end{bmatrix}+\frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}+\frac{1}{3}\begin{bmatrix}1\\0\\1\end{bmatrix}=$$

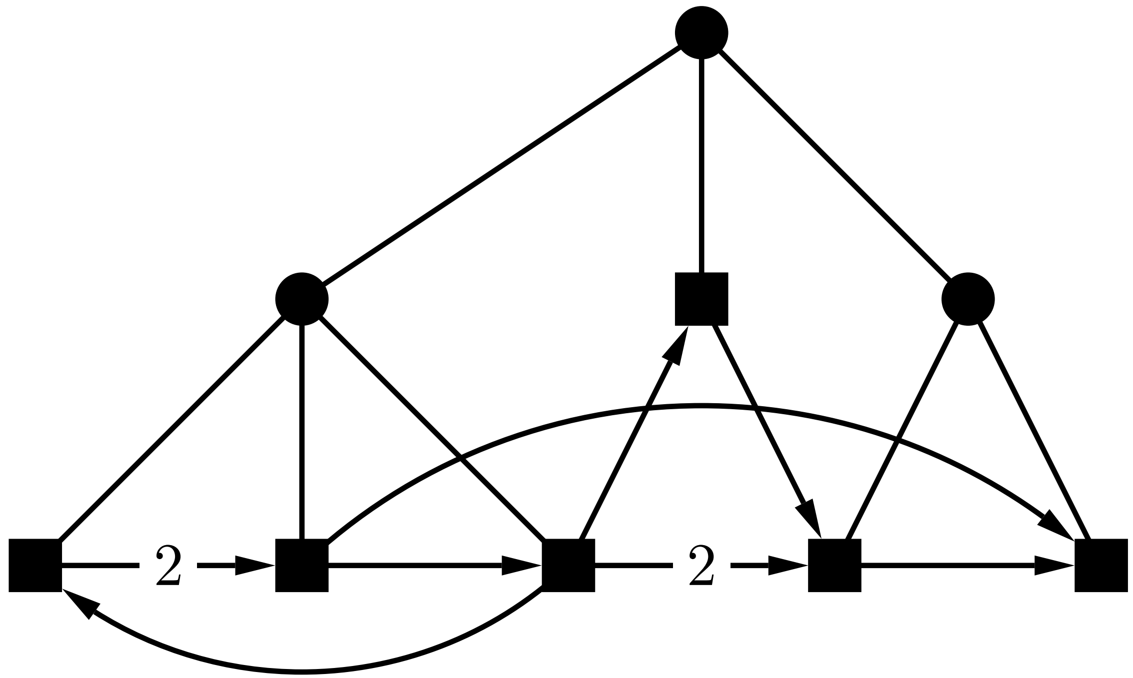
$$\begin{bmatrix}1&1&1\\1&1&0\\2&1&1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix}/3=\begin{bmatrix}3\\2\\4\end{bmatrix}/3\rightarrow\begin{bmatrix}\frac{3}{11}\\\frac{2}{11}\\\frac{4}{11}\end{bmatrix}$$

- This generalizes Indegree.

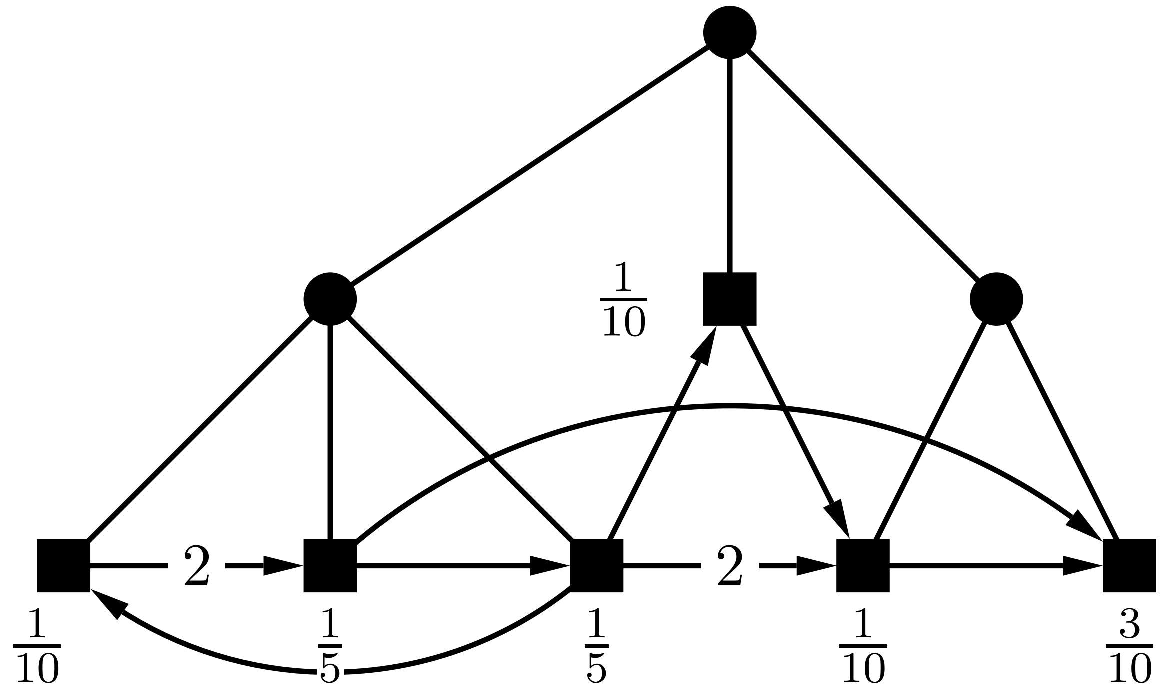
Ranking and Hierarchies



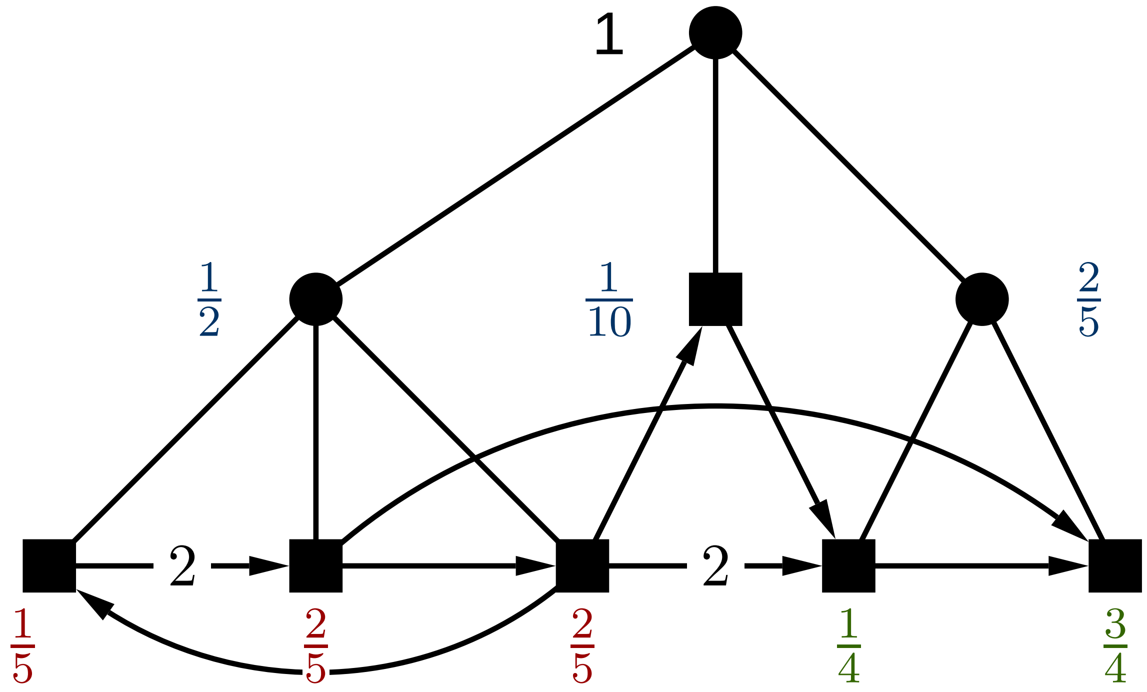
Ranking and Hierarchies



Ranking and Hierarchies



Ranking and Hierarchies



Ranking and Hierarchies

Peer-review Principle

Ranking and Hierarchies

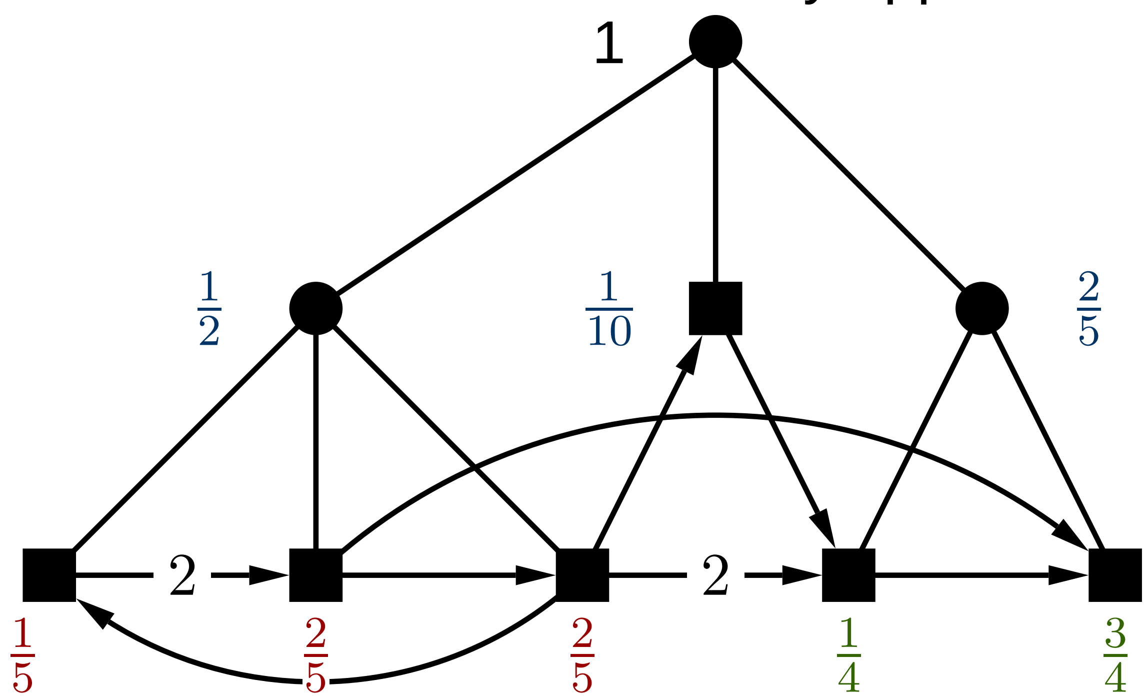
Peer-review Principle

Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.

Ranking and Hierarchies

Peer-review Principle

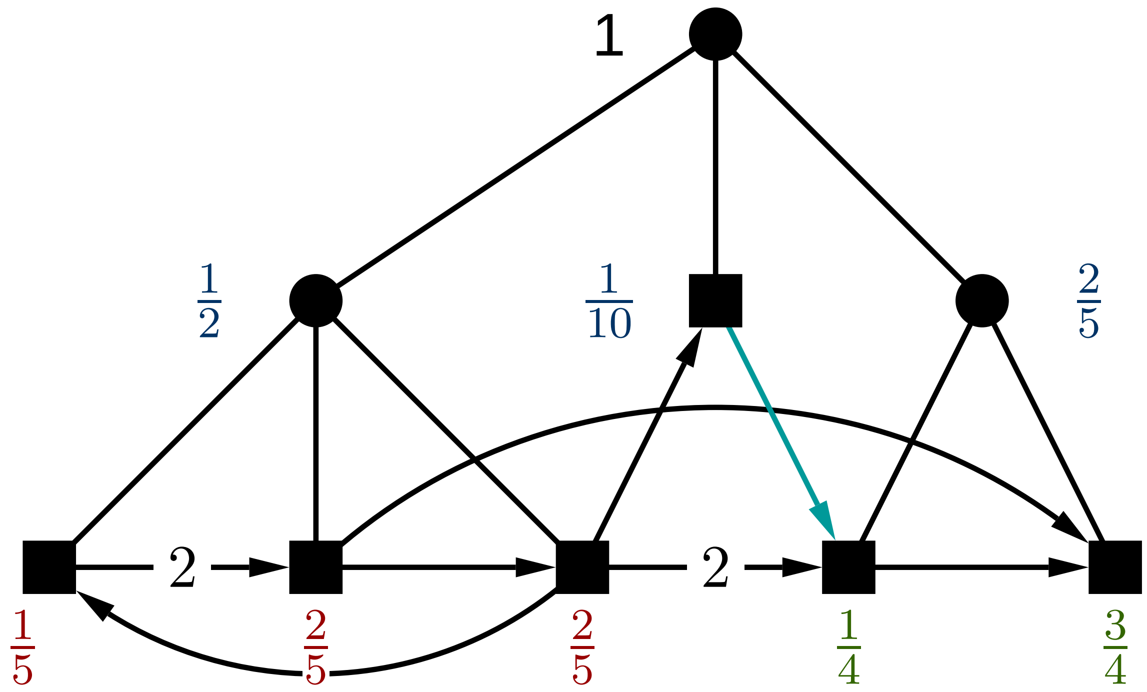
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

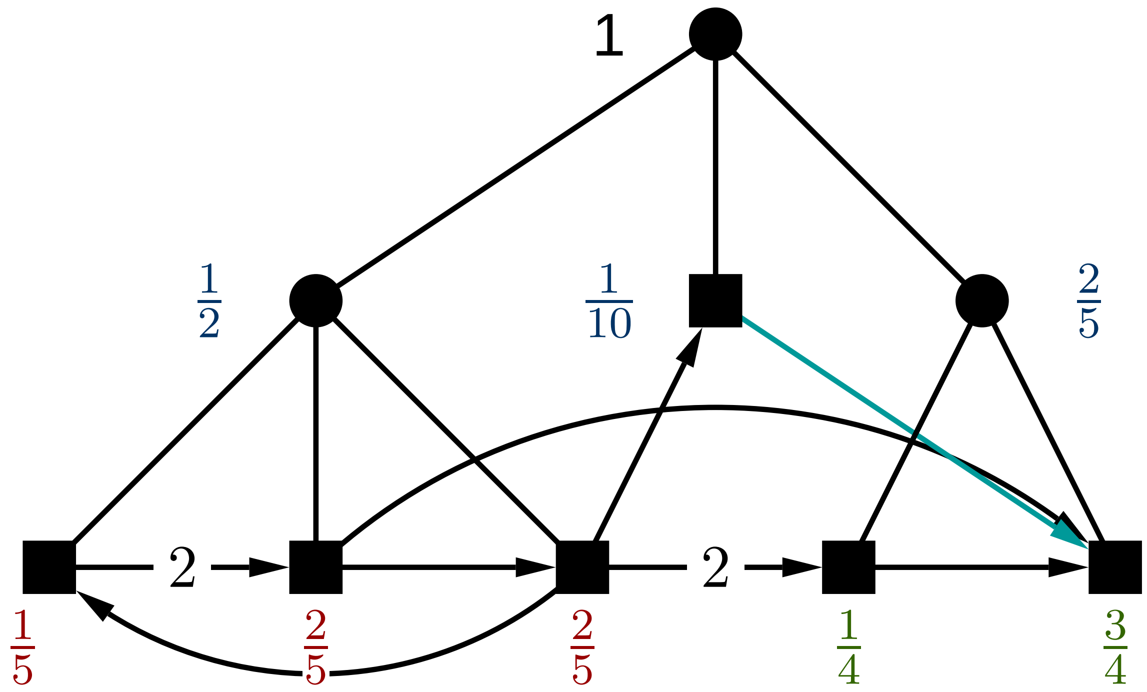
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

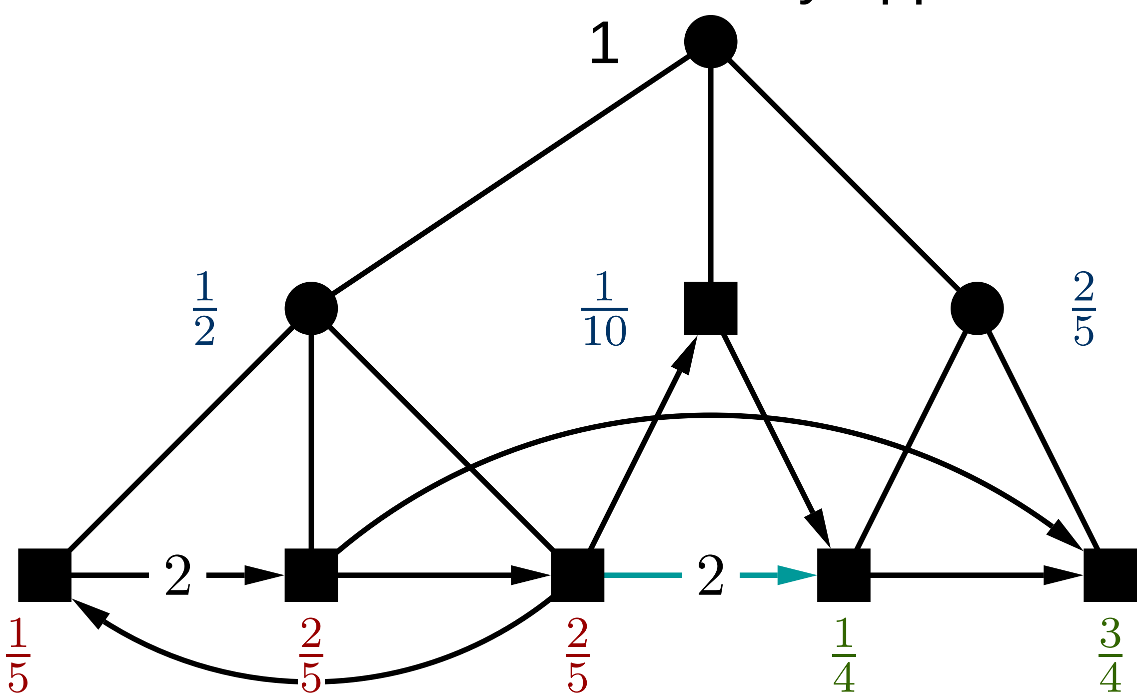
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

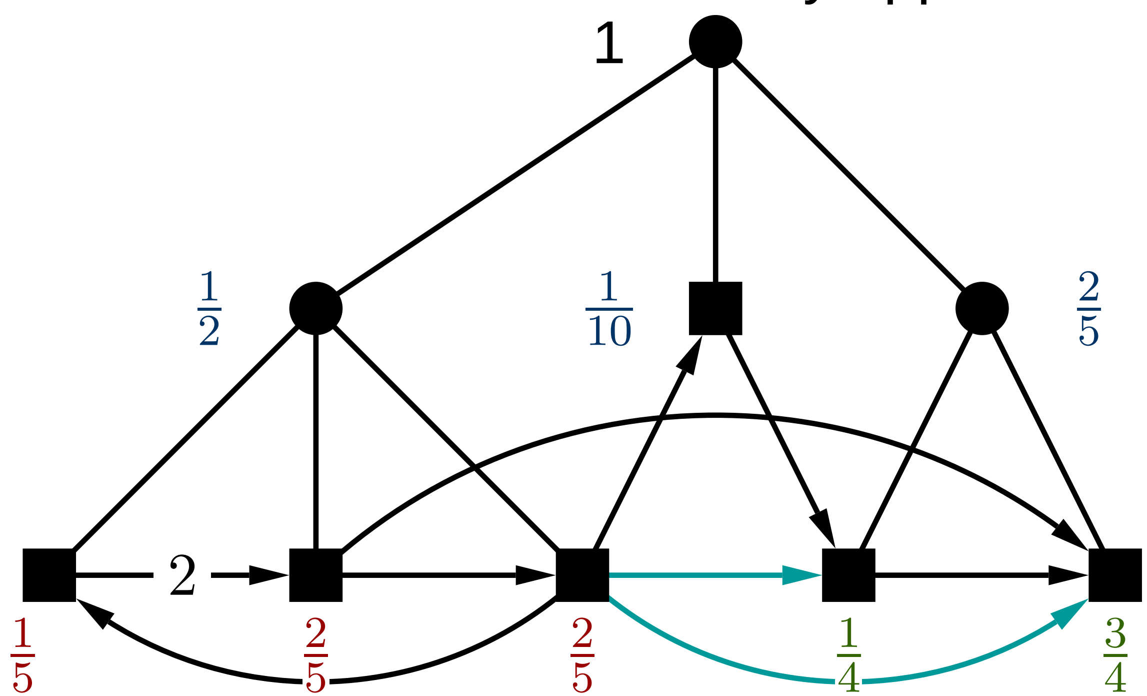
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

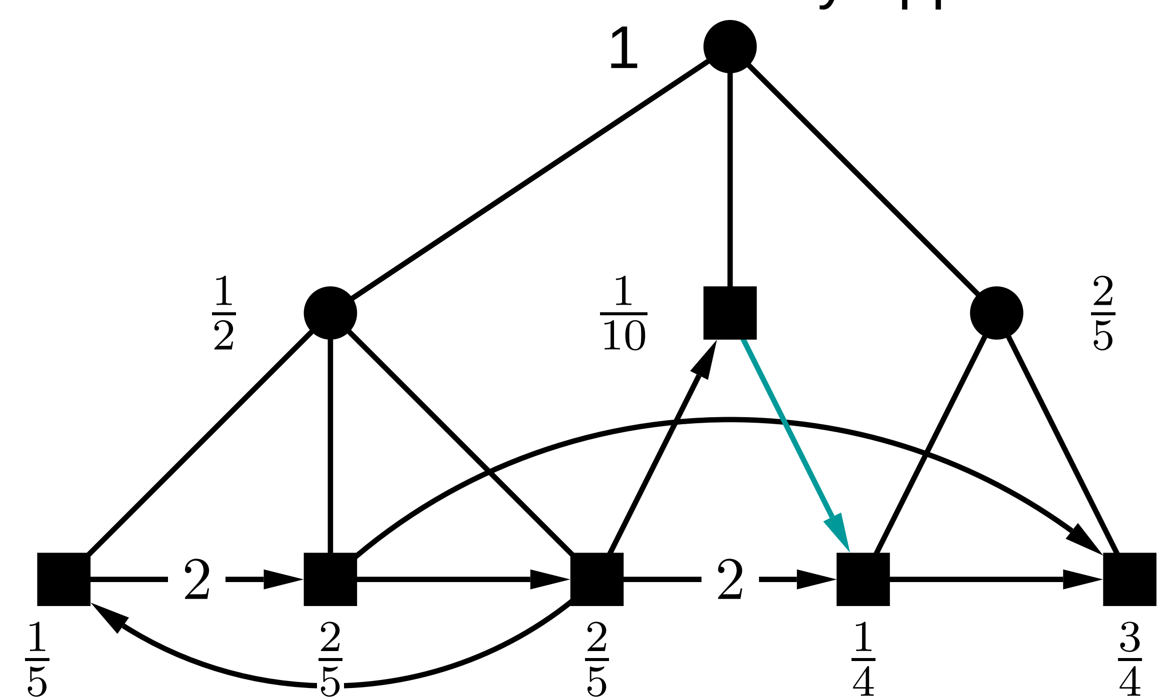
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

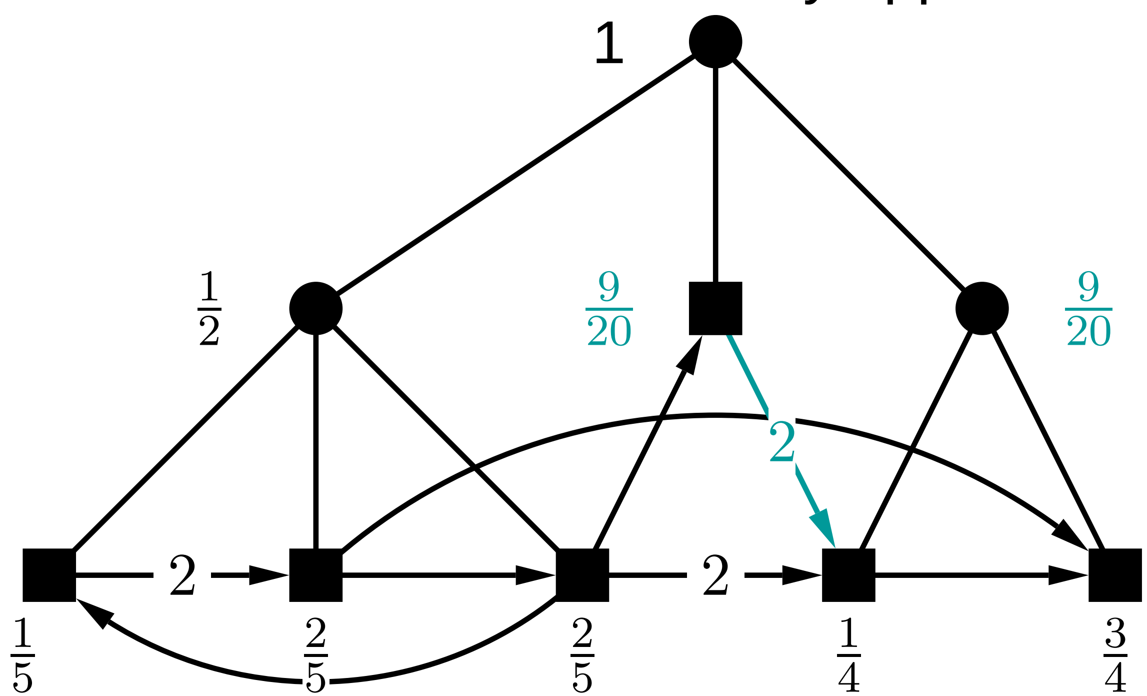
Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



Ranking and Hierarchies

Peer-review Principle

Endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate.



QuickRank

QuickRank enforces this peer-review principle by:

QuickRank

QuickRank enforces this peer-review principle by:

- recursively applying any given “base” ranking algorithm to update a given prior ranking,

QuickRank

QuickRank enforces this peer-review principle by:

- recursively applying any given “base” ranking algorithm to update a given prior ranking,
- exploiting Bonacich’s hypothesis to aggregate importance judgments.

QuickRank

QuickRank enforces this peer-review principle by:

- recursively applying any given “base” ranking algorithm to update a given prior ranking,
- exploiting Bonacich’s hypothesis to aggregate importance judgments.

As such, it is a *meta*-ranking algorithm:

QuickRank

QuickRank enforces this peer-review principle by:

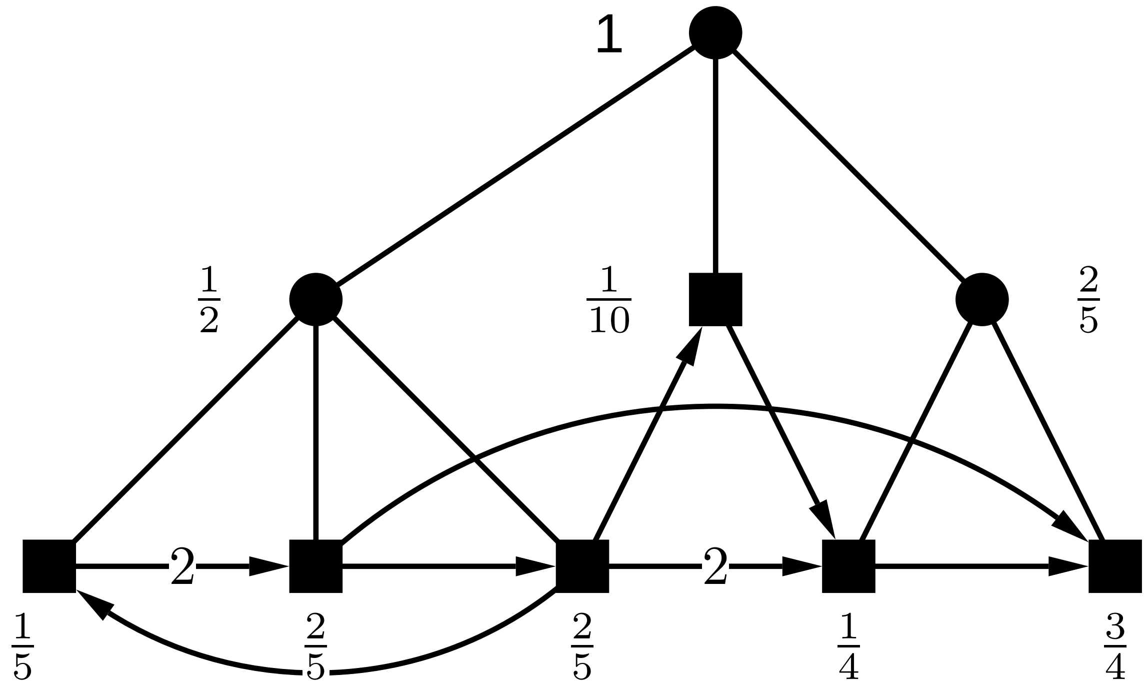
- recursively applying any given "base" ranking algorithm to update a given prior ranking,
- exploiting Bonacich's hypothesis to aggregate importance judgments.

As such, it is a *meta*-ranking algorithm:

$$\text{Ranking Algorithm} \times \text{Hierarchy} \longrightarrow \text{Ranking Algorithm}$$

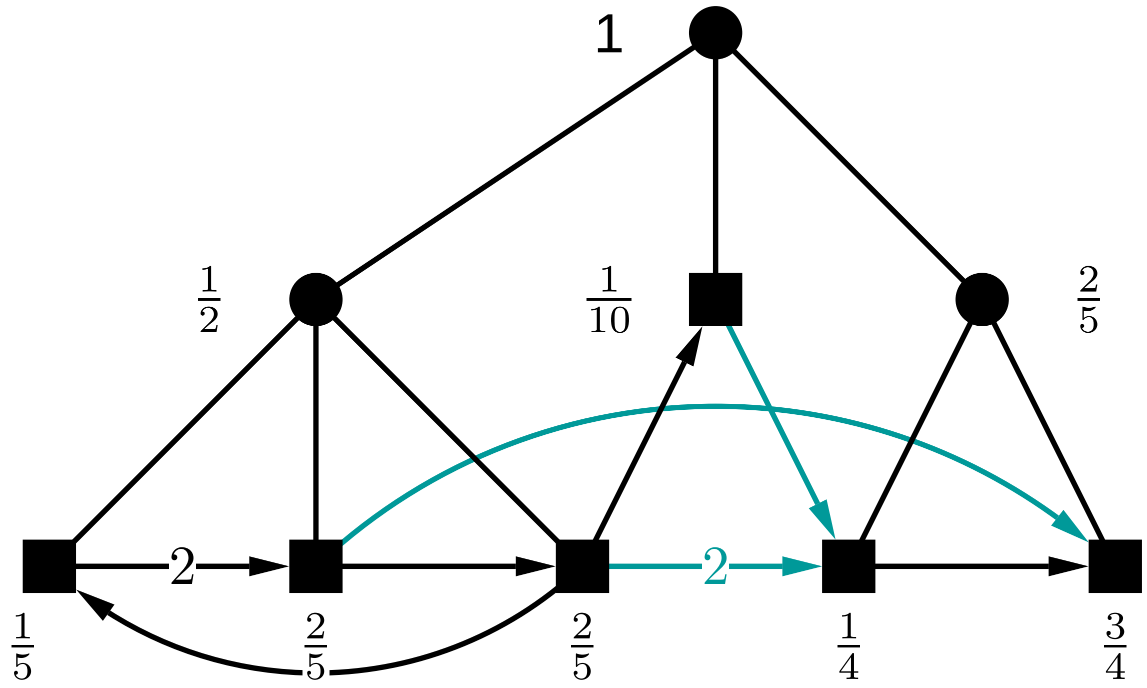
satisfying Peer-Review

QuickRank: A Sample Calculation



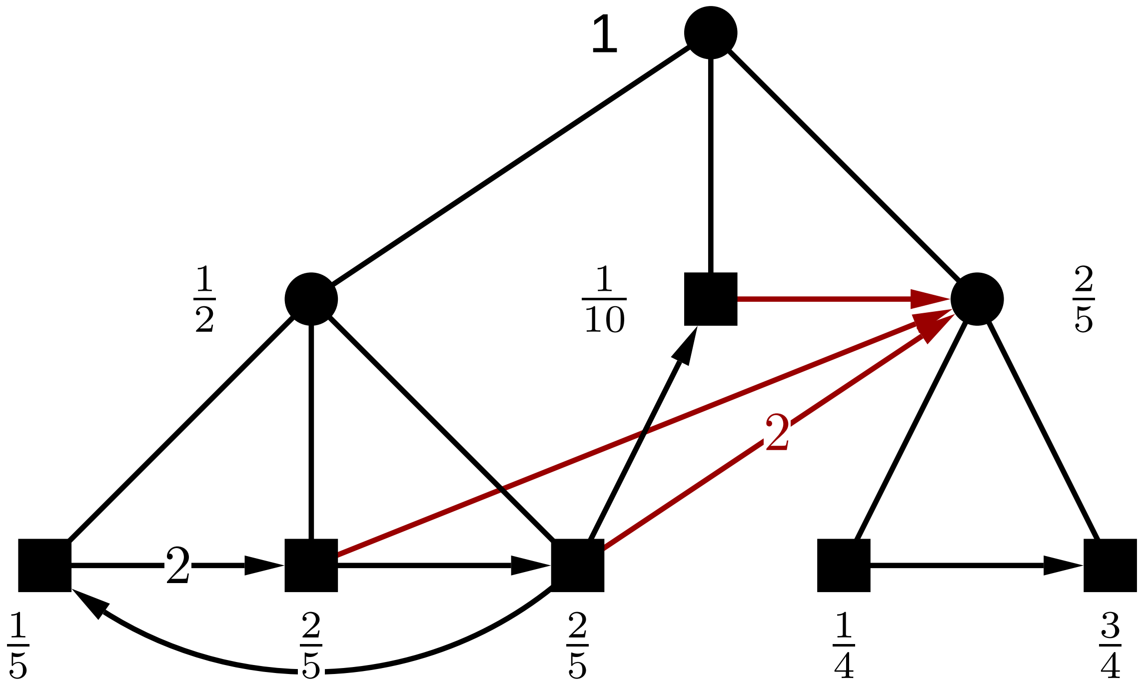
Apply the Peer-Review Principle to localize the importance judgments.

QuickRank: A Sample Calculation



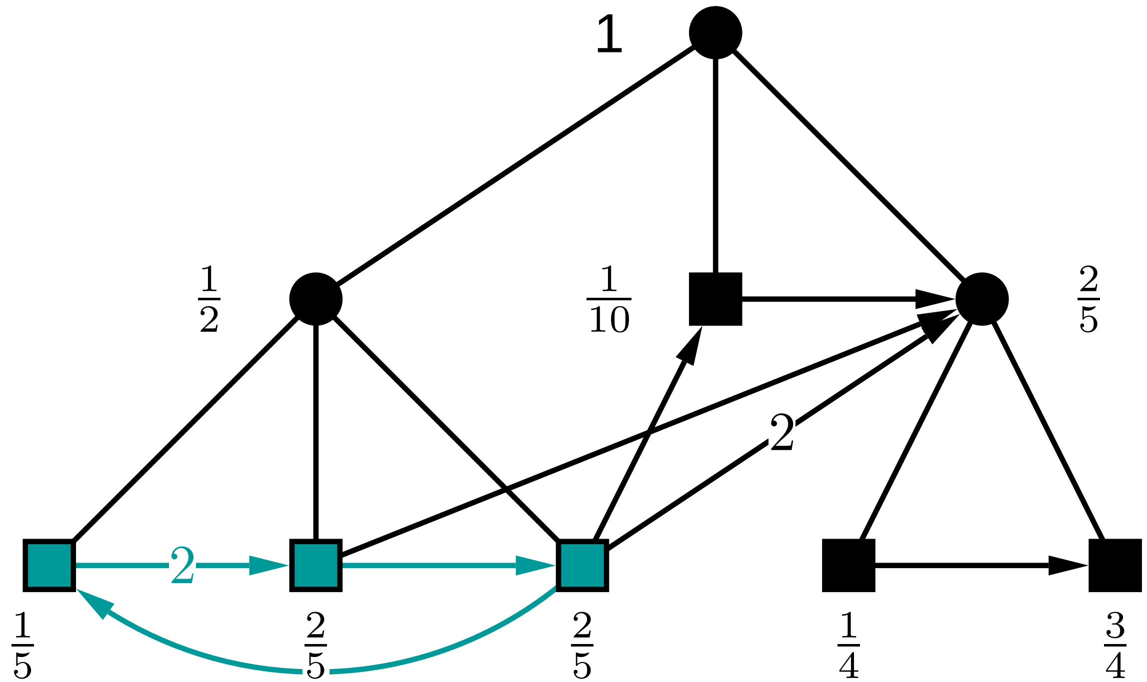
Apply the Peer-Review Principle to localize the importance judgments.

QuickRank: A Sample Calculation



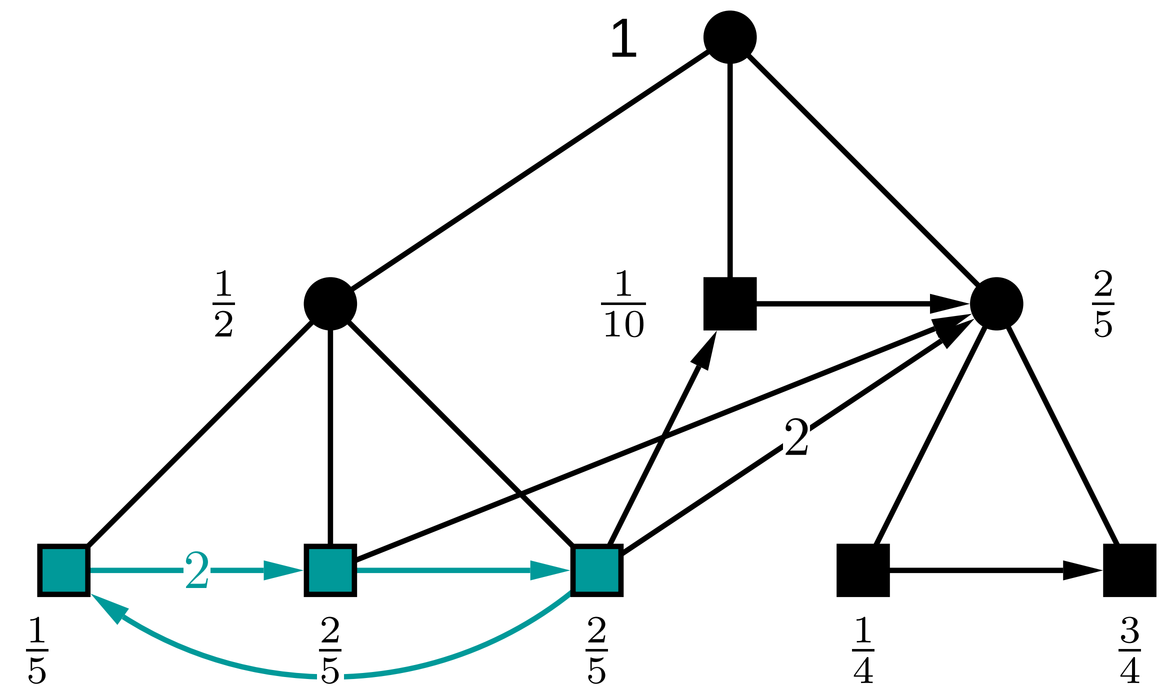
Apply the Peer-Review Principle to localize the importance judgments.

QuickRank: A Sample Calculation



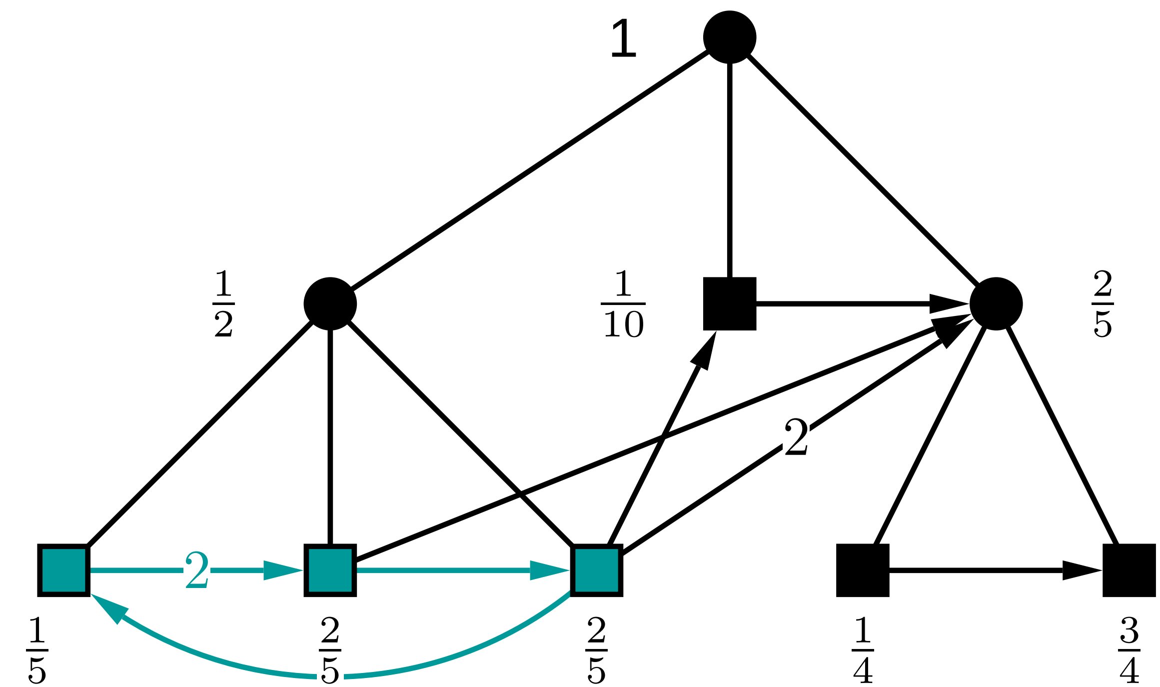
Apply the Peer-Review Principle to focus on a particular subcommunity.

QuickRank: A Sample Calculation



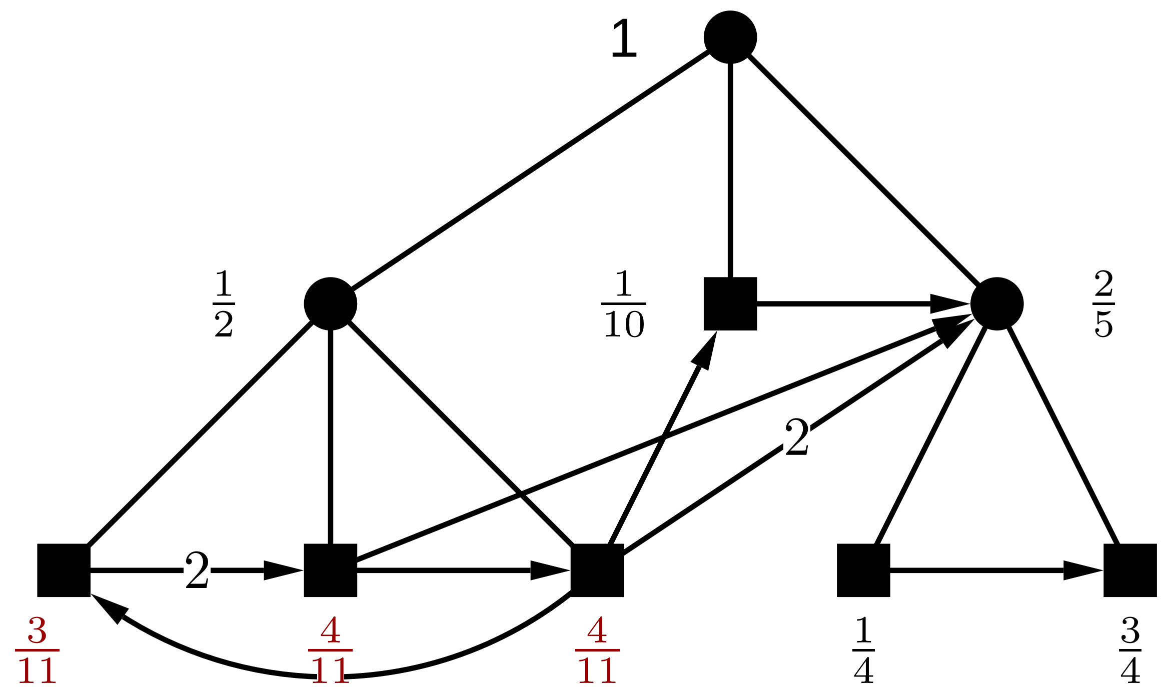
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} \\ \frac{2}{5} \\ \frac{2}{5} \end{bmatrix}$

QuickRank: A Sample Calculation



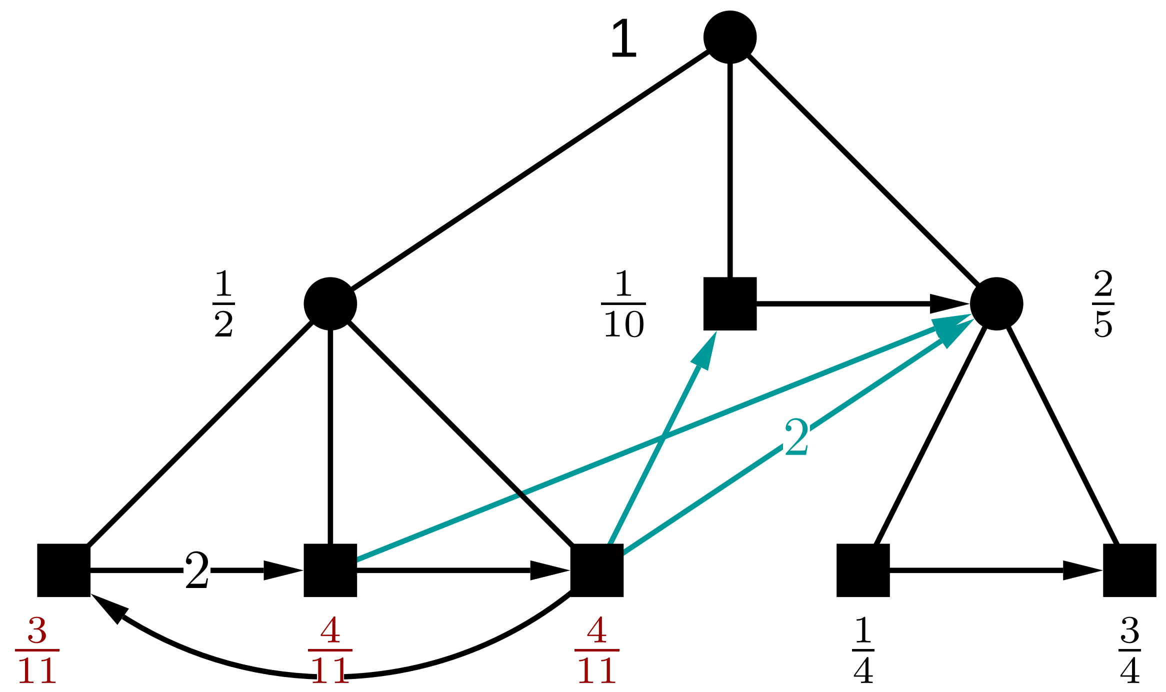
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} \\ \frac{2}{5} \\ \frac{2}{5} \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \\ \frac{4}{5} \end{bmatrix}$

QuickRank: A Sample Calculation



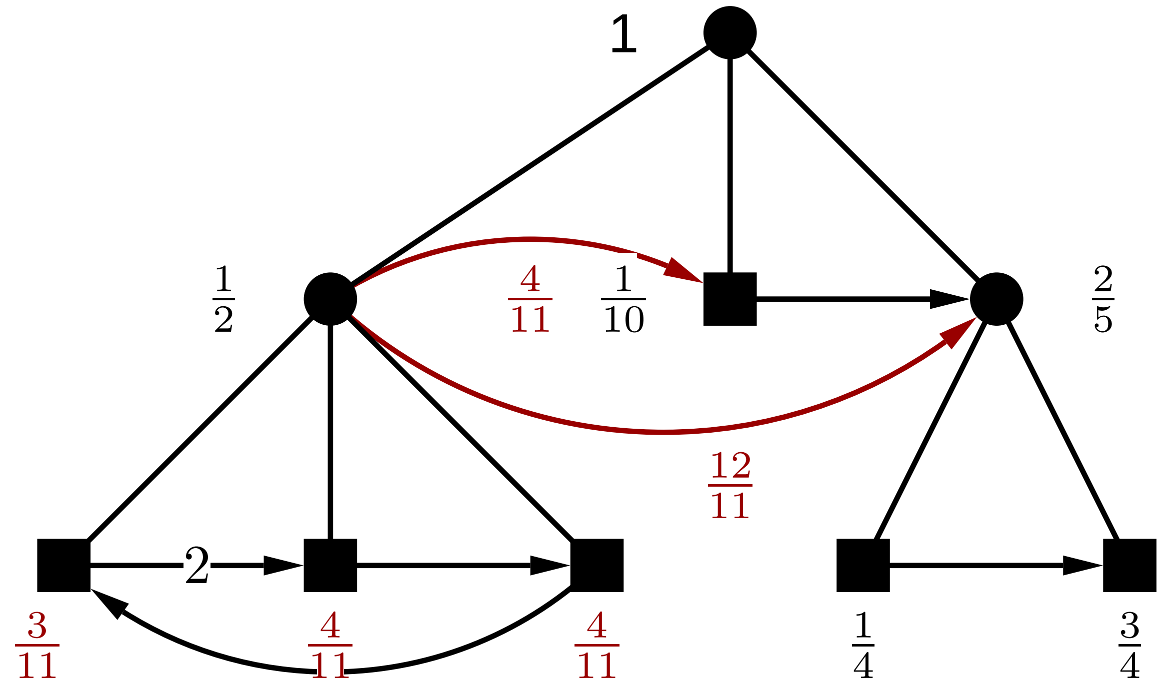
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} \\ \frac{2}{5} \\ \frac{2}{5} \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \\ \frac{4}{5} \end{bmatrix}$

QuickRank: A Sample Calculation



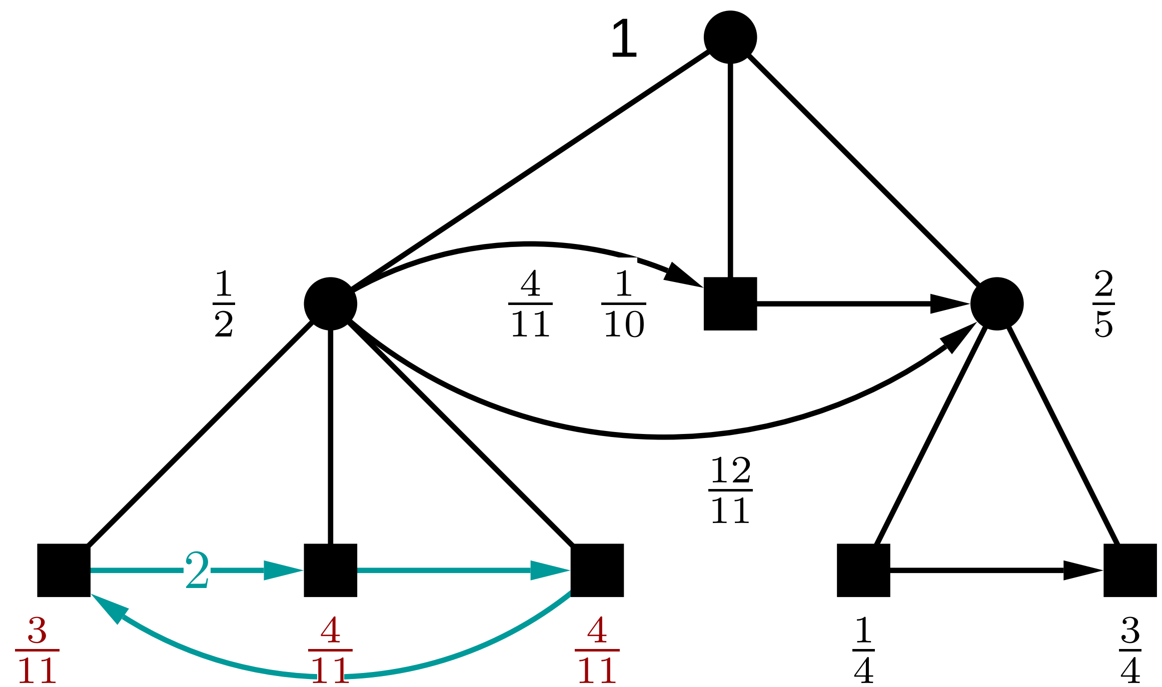
Apply Bonacich's Hypothesis to aggregate non-local judgments.

QuickRank: A Sample Calculation



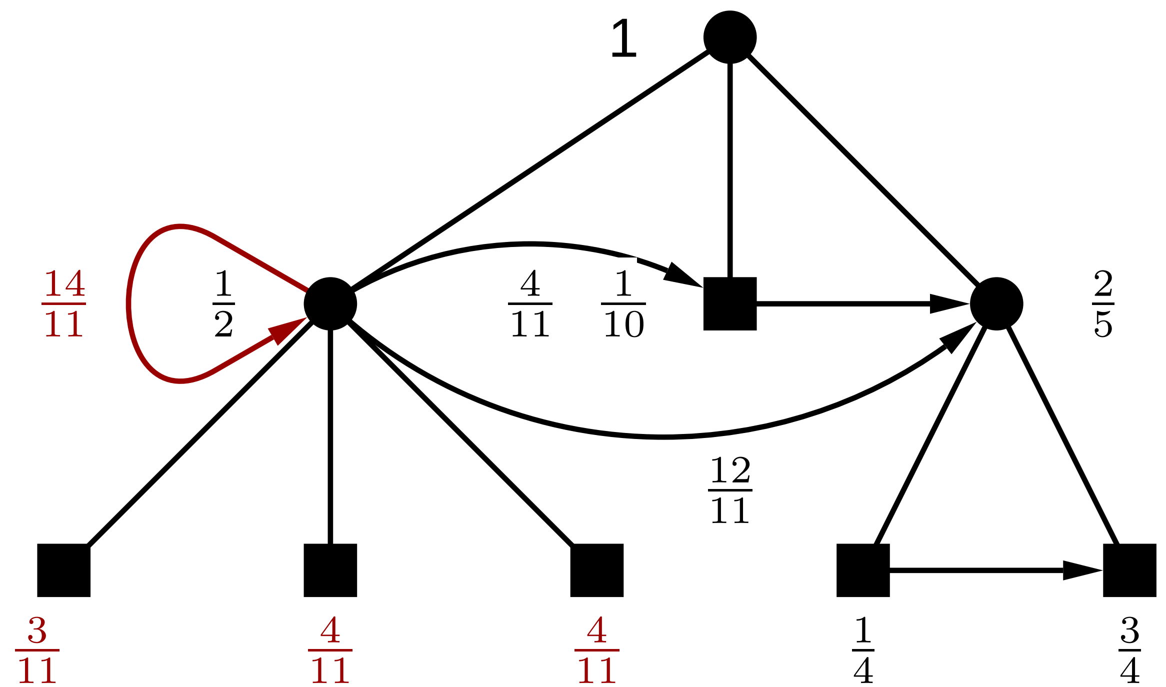
Apply Bonacich's Hypothesis to aggregate non-local judgments.

QuickRank: A Sample Calculation



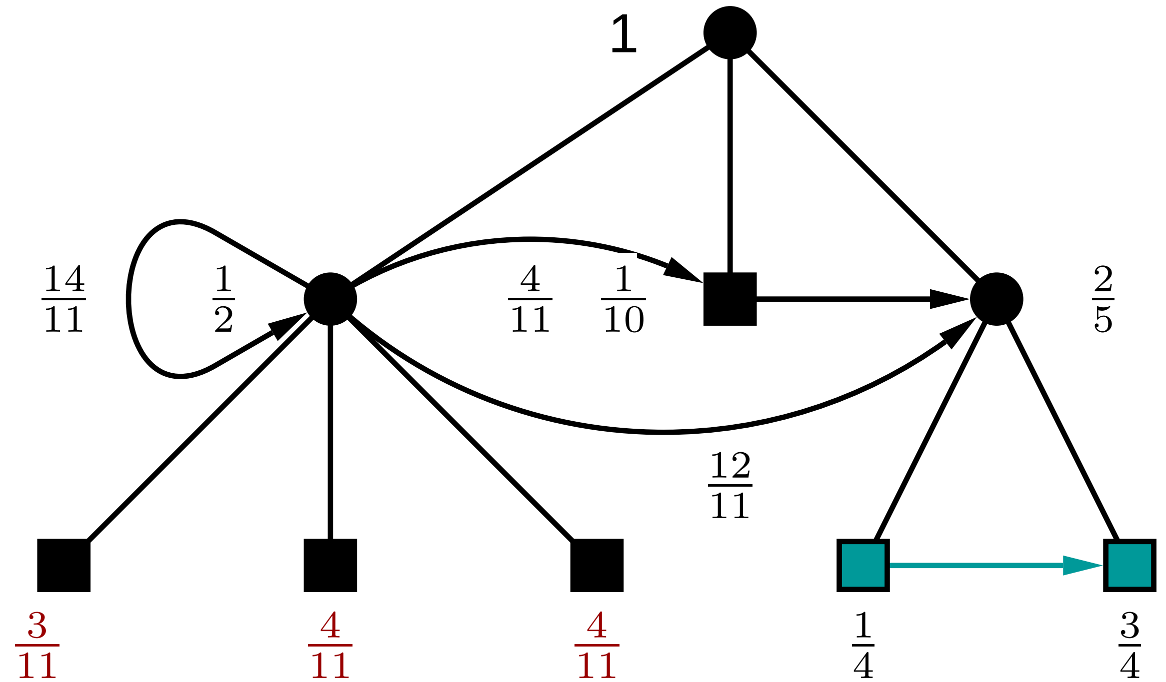
Apply Bonacich's Hypothesis and the Peer-Review Principle to aggregate local judgments: $2 \cdot \frac{3}{11} + \frac{4}{11} + \frac{4}{11}$

QuickRank: A Sample Calculation



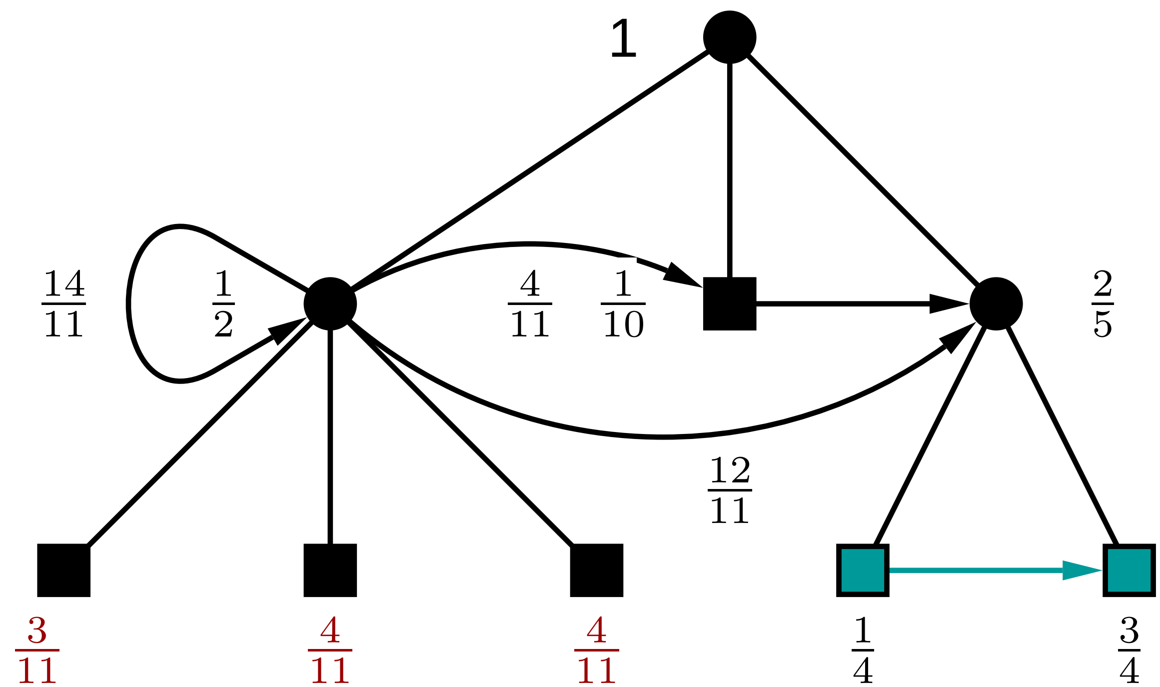
Apply Bonacich's Hypothesis and the Peer-Review Principle to aggregate local judgments: $2 \cdot \frac{3}{11} + \frac{4}{11} + \frac{4}{11} = \frac{14}{11}$

QuickRank: A Sample Calculation



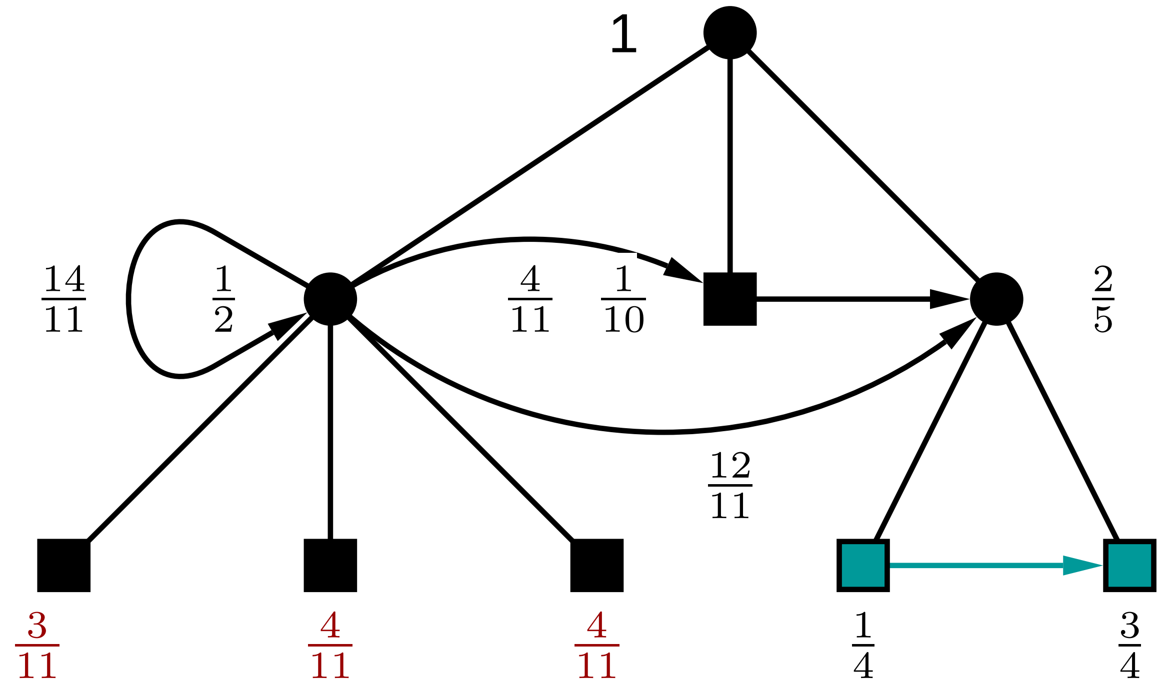
Apply the Peer-Review Principle to focus on a particular subcommunity.

QuickRank: A Sample Calculation



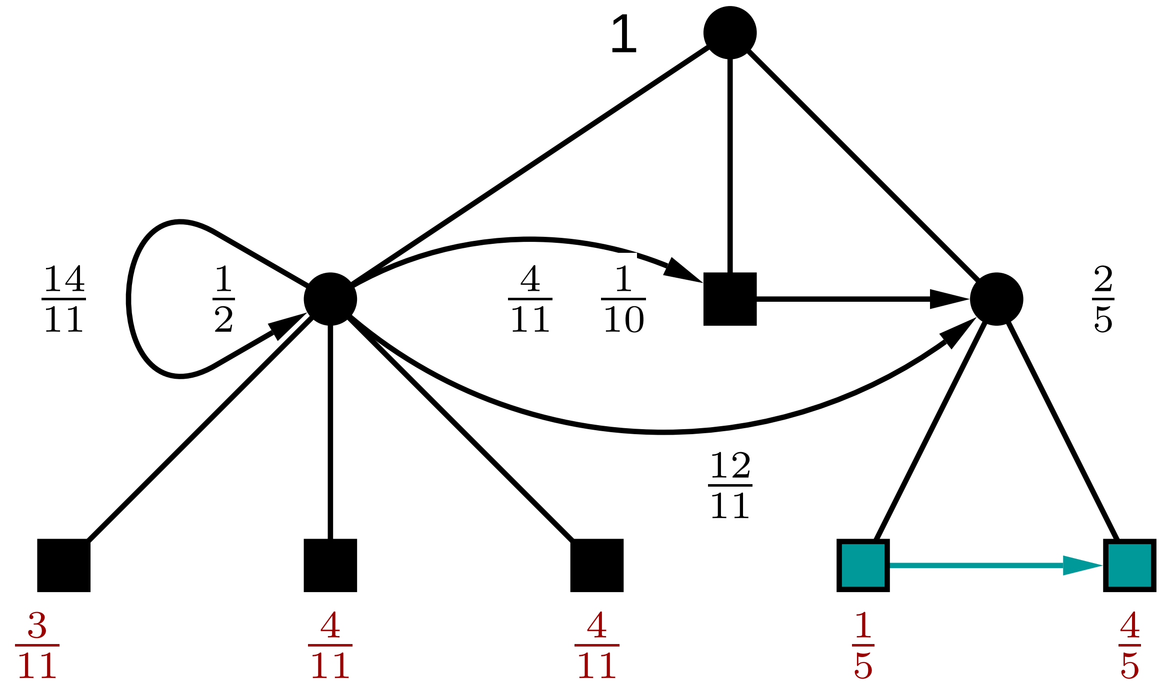
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix}$

QuickRank: A Sample Calculation



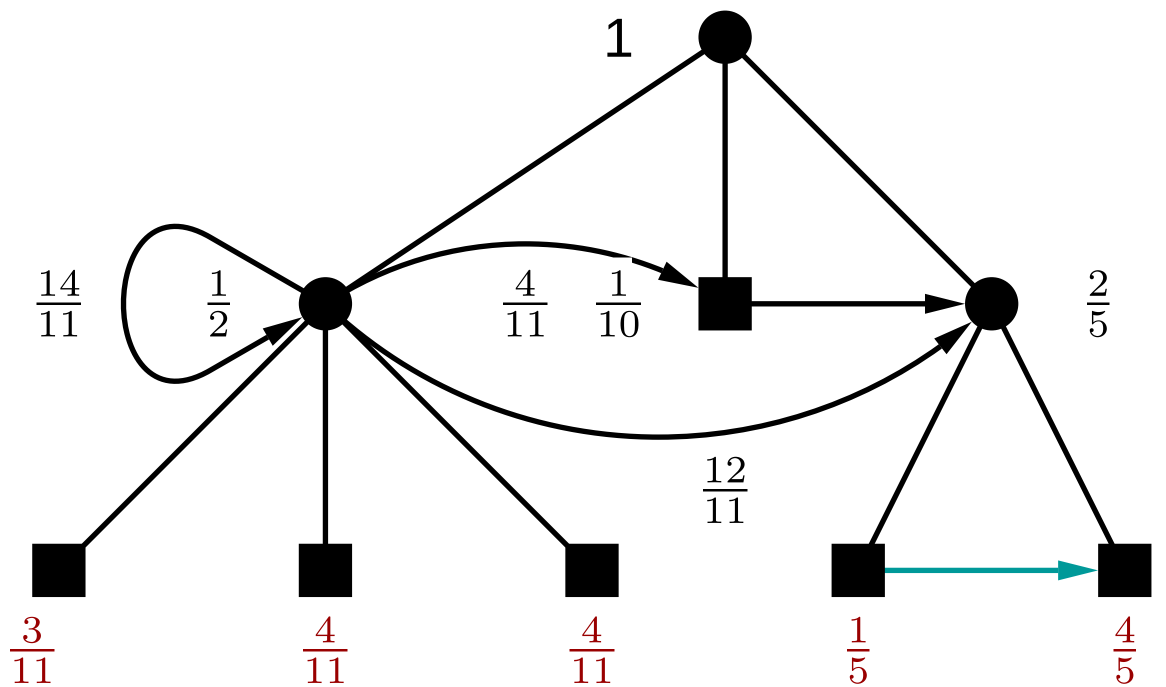
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ 1 \end{bmatrix}$

QuickRank: A Sample Calculation



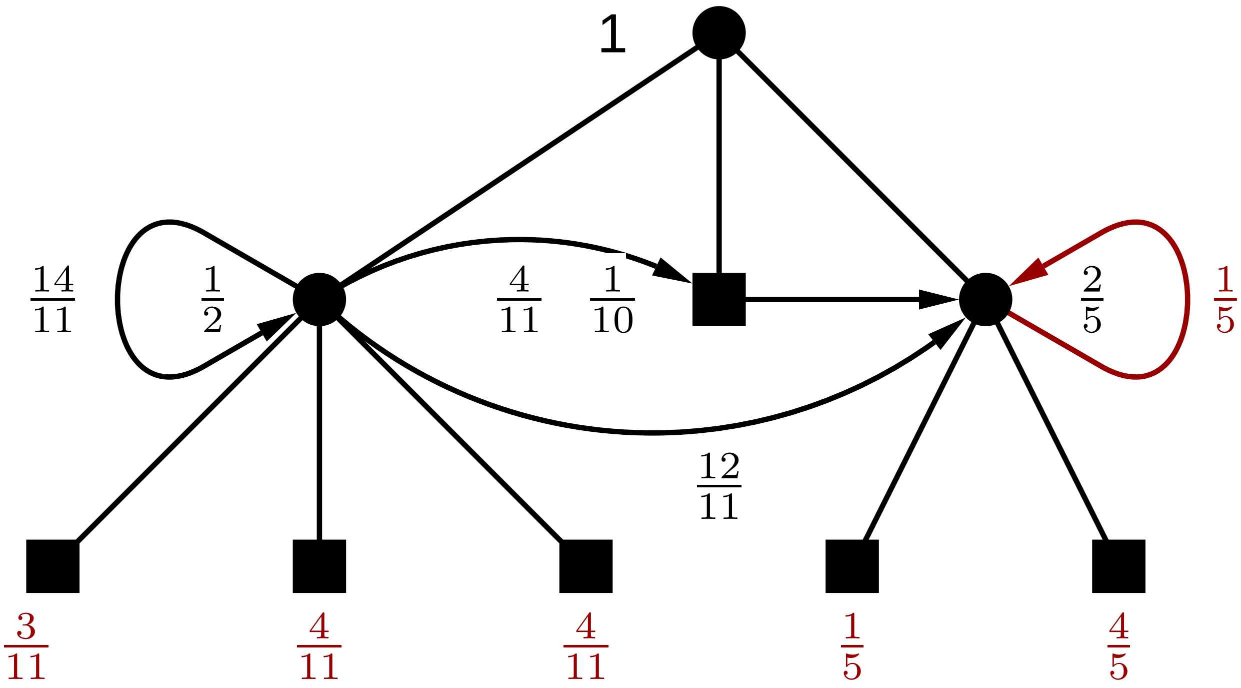
Apply Indegree to rank the subcommunity: $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ 1 \end{bmatrix}$

QuickRank: A Sample Calculation



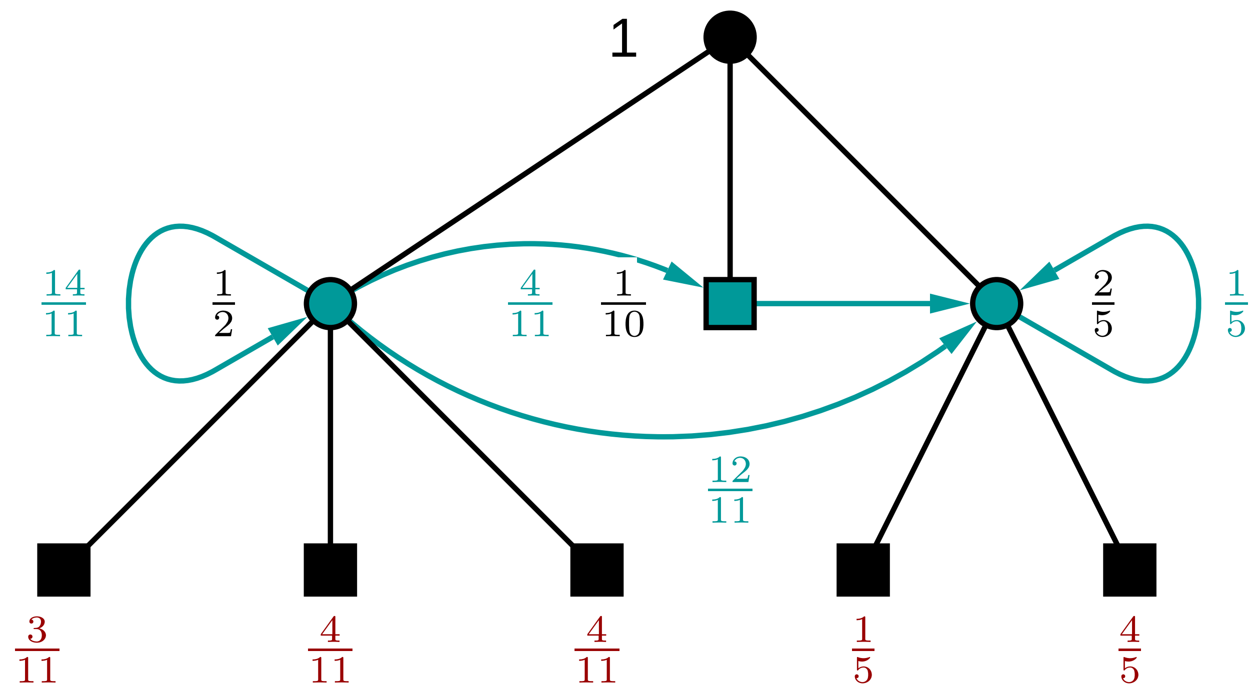
Apply Bonacich's Hypothesis and the Peer-Review Principle to aggregate local judgments.

QuickRank: A Sample Calculation



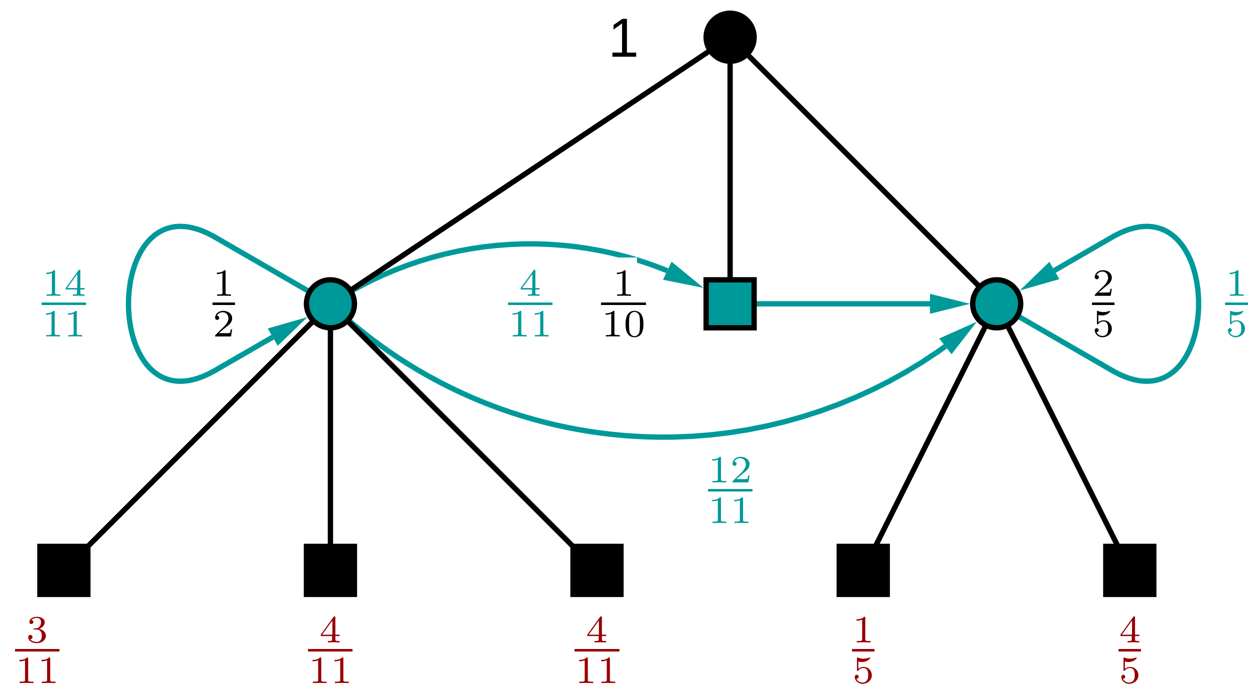
Apply Bonacich's Hypothesis and the Peer-Review Principle to aggregate local judgments.

QuickRank: A Sample Calculation



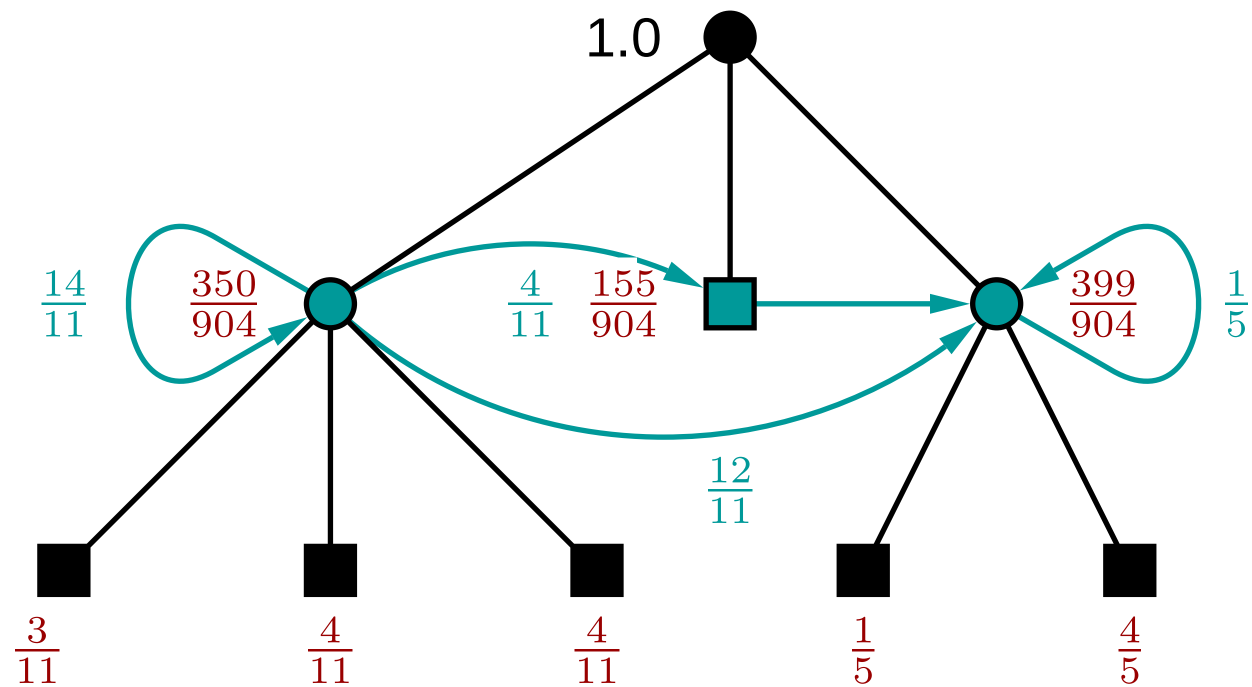
Apply the Peer-Review Principle to focus on a particular subcommunity.

QuickRank: A Sample Calculation



Apply Indegree to rank the subcommunity: $\begin{bmatrix} \frac{14}{11} & 0 & 0 \\ \frac{4}{11} & 1 & 0 \\ \frac{12}{11} & 1 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.1 \\ 0.4 \end{bmatrix}$

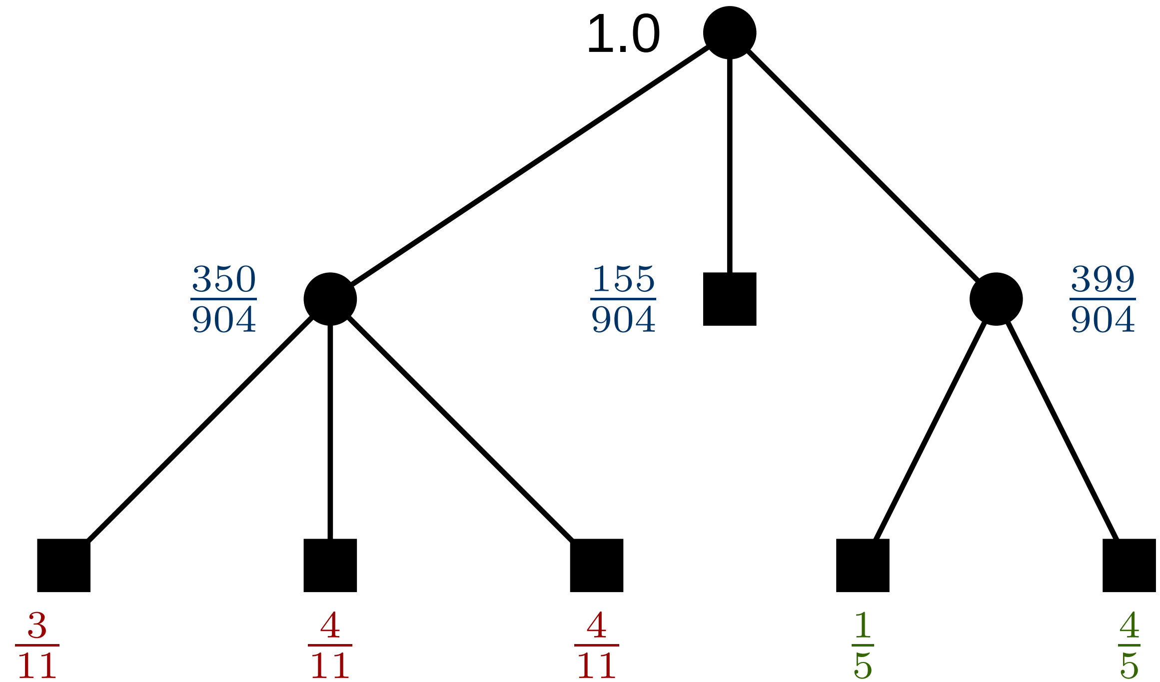
QuickRank: A Sample Calculation



Apply Indegree to rank the subcommunity:

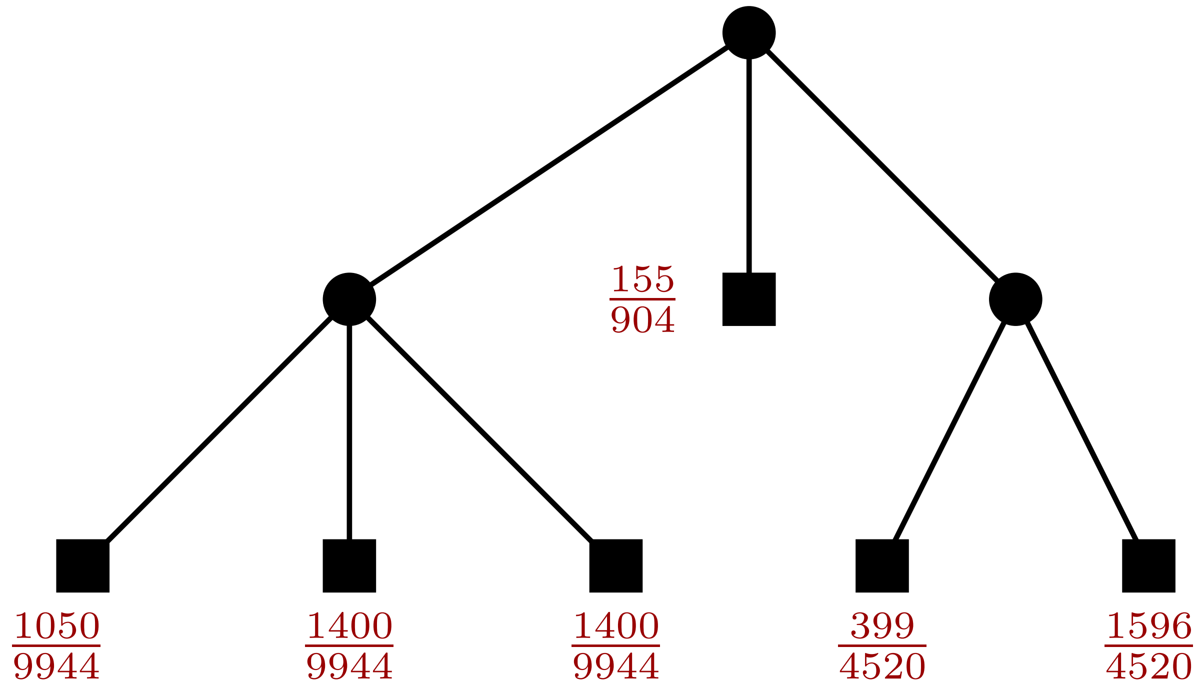
$$\begin{bmatrix} \frac{14}{11} & 0 & 0 \\ \frac{4}{11} & 1 & 0 \\ \frac{12}{11} & 1 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.1 \\ 0.4 \end{bmatrix} = \begin{bmatrix} \frac{7}{11} \\ \frac{31}{110} \\ \frac{399}{550} \end{bmatrix}$$

QuickRank: A Sample Calculation



Collapse rankings.

QuickRank: A Sample Calculation



Collapse ranking.

Comparison with TREC 2003

P@10	AP	P@R	α	Alg	Depth
0.124	0.154	0.164	-	csiro03td03	-
0.090	0.099	0.114	0.97	Indegree	1
0.086	0.097	0.105	0.97	Indegree	0
0.082	0.089	0.086	1.00	Lucene	-
0.074	0.088	0.092	0.97	PageRank	0
0.062	0.087	0.078	0.97	PageRank	1
0.092	0.070	0.092	-	meijihilw1	-
0.032	0.023	0.028	-	C2B	-

Comparison with TREC 2004

S@1	S@5	S@10	P@10	R@M	AP	α	Alg	Depth
0.507	0.773	0.893	0.249	0.777	0.179	-	uogWebCAU150	-
0.213	0.680	0.773	0.151	0.590	0.123	0.95	Indegree	1
0.253	0.680	0.813	0.163	0.590	0.120	0.95	Indegree	0
0.333	0.64	0.76	0.199	0.647	0.115	-	MU04web1	-
0.227	0.587	0.707	0.135	0.586	0.093	0.95	PageRank	0
0.080	0.400	0.573	0.109	0.569	0.075	1.00	Lucene	-
0.187	0.533	0.600	0.097	0.582	0.074	0.95	PageRank	1
0.067	0.147	0.173	0.029	0.147	0.018	-	irttil	-

Desdirata for Ranking Algorithms

- **Consensus**

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking,

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus,

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus, independent of prior ranking.

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus, independent of prior ranking.

- **Spam Resistance**

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus, independent of prior ranking.

- **Spam Resistance**

  If *sybils* are introduced into the community,

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus, independent of prior ranking.

- **Spam Resistance**

  If *sybils* are introduced into the community, but are given 0 prior ranking,

Desdirata for Ranking Algorithms

- **Consensus**

  If all individual importance judgments express a common ranking, the posterior ranking should always equal this consensus, independent of prior ranking.

- **Spam Resistance**

  If *sybils* are introduced into the community, but are given 0 prior ranking, the posterior ranking of the original members should be unaffected.

Properties of QuickRank

Preserves:

- consensus
- spam-resistance
- identity

Future Work

- Applications

Examples of HSN's	Individuals	Network	Hierarchy
the Web	web pages	hyperlinks	domains, subdomains, etc.
citation index	publications	references	fields, subfields, etc.
the Enron email DB	employees	emails	organizational chart
Suggestions	?	?	?

Future Work

- Applications

Examples of HSN's	Individuals	Network	Hierarchy
the Web	web pages	hyperlinks	domains, subdomains, etc.
citation index	publications	references	fields, subfields, etc.
the Enron email DB	employees	emails	organizational chart
Suggestions	?	?	?

- Generalization to weighted DAG's.

Thanks