Mathematisches Konzept: Markov-Ketten

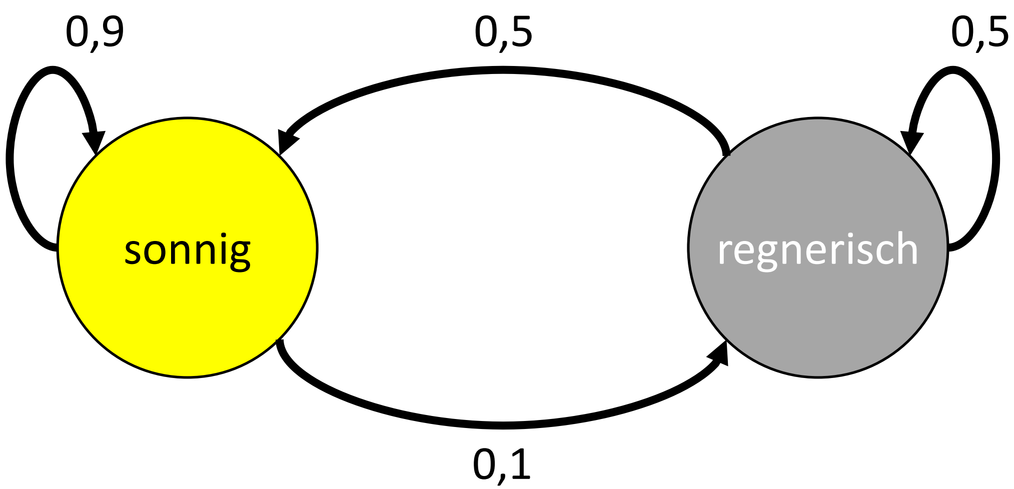
- Eine **Markov-Kette** ist ein spezieller stochastischer Prozess, der **Zustände** und die **Wahrscheinlichkeit ihrer Übergänge** beschreibt.
- Zustände aus Menge S (bei uns: mögliche Webseiten, daher **endlich**)
- Beobachtung zu diskreten Zeitpunkten aus $T = \{0,1,2, \dots\}$
- Zustand im Zeitpunkt t ist Zufallsvariable X_t mit Werten aus S
- Wir betrachten nur **Markovketten erster Ordnung**, d.h. es gilt für die Übergangswahrscheinlichkeiten

$$P[X_t = x | X_{t-1} = w, \dots, X_0 = a] = P[X_t = x | X_{t-1} = w]$$

  Der nächste Zustand hängt also nur vom aktuellen Zustand ab, die Kette ist **gedächtnislos**.
- Die Übergangswahrscheinlichkeiten ändern sich nicht mit der Zeit, d.h. unsere Markovketten sind **zeitinvariant**.

Beispiel: Markov-Kette

Wie wird das Wetter morgen sein, wenn es heute sonnig ist?



Die Wahrscheinlichkeit, dass es morgen regnet, wenn es heute sonnig ist, ist 0,1

$$P[X_t = regnerisch | X_{t-1} = sonnig] = 0.1$$

Dann ist es mit dieser Wahrscheinlichkeit im nächsten Schritt regnerisch

Darstellung der Kette als **Übergangsmatrix**:

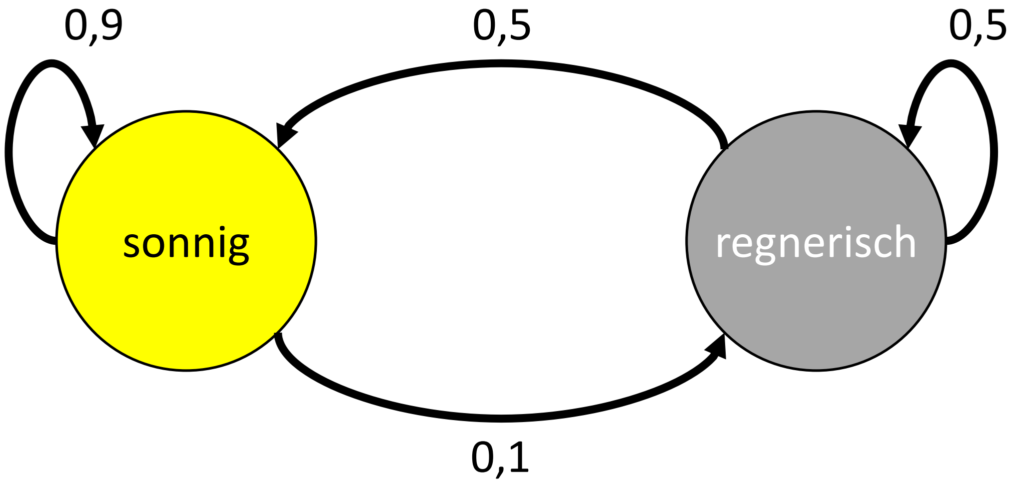
Wenn es jetzt sonnig ist

$$\boldsymbol{P} = \begin{pmatrix} 0{,}9 & 0{,}5 \\ \boxed{0{,}1} & 0{,}5 \end{pmatrix}$$

Spaltensumme immer 1 (**spaltenstochastisch**)

P_{ki} ist die Wahrsch., dass der folgende Tag vom Typ k ist, wenn der heutige Tag vom Typ i ist.

Beispiel: Markov-Kette



$$\boldsymbol{P} = \begin{pmatrix} 0{,}9 & 0{,}5 \\ 0{,}1 & 0{,}5 \end{pmatrix}$$

Der Vektor $\boldsymbol{x}^{(t)}$ beschreibt die Wahrscheinlichkeiten, mit der die einzelnen Zustände in Schritt t angenommen werden.

Nehmen wir an, dass es initial sonnig ist, also $\boldsymbol{x}^{(\mathbf{0})} = (1 \quad 0)$. Dann gilt für die Wahrscheinlichkeiten im nächsten Schritt

$$\boldsymbol{x}^{(\mathbf{1})} = \boldsymbol{P}\boldsymbol{x}^{(\mathbf{0})} = \begin{pmatrix} 0{,}9 & 0{,}5 \\ 0{,}1 & 0{,}5 \end{pmatrix} (1 \quad 0) = (0{,}9 \quad 0{,}1)$$

Wir interessieren uns für die **stationäre Zustandsverteilung**

$$\boldsymbol{\pi} = \lim_{\boldsymbol{t} \to \infty} \boldsymbol{x}^{(\boldsymbol{t})}$$

(existiert unter bestimmten Bedingungen, die hier und für PageRank zutreffen.)

Suche mit PageRank

Zur Suche nach Webseiten über PageRank gibt es 2 Ansätze:

Ansatz 1: **PageRank für Sortierung**

1. Suche alle zur Suchanfrage passenden Webseiten anhand eines IR-Verfahrens
2. Ordne die Webseiten in absteigender Reihenfolge entsprechend ihres PageRanks

Ansatz 2: **PageRank für Relevanzbestimmung**

1. Kombiniere PageRank mit einem IR-Verfahren zur Relevanzschätzung von Webseiten
2. Ordne die Webseiten in absteigender Reihenfolge entsprechend ihrer Relevanz

Suche mit PageRank

Der 1. Ansatz zur Suche mit PageRank ist einfach zu implementieren. Die Sortierung der Webseiten erfolgt aufgrund ihrer Wichtigkeit im Webspace **unabhängig von deren geschätzter Relevanz** für die Suchanfrage.

Der 2. Ansatz erfordert eine **komplexe Kombination von PageRank und IR-Scoring**. Die Sortierung der Webseiten erfolgt aufgrund ihrer Wichtigkeit im Webspace und unter Berücksichtigung von deren Relevanz für die Suchanfrage.

In der Praxis verwenden Suchmaschinen eine Variante des 2. Ansatzes.

Websuchmaschinen

- Ranking mit HITS

Webmodell

Der **Webspace W** ist ein gerichteter Graph $G = (V,E)$. Die Knoten V repräsentieren Webseiten im Webspace. Die Kanten E entsprechen der Verlinkung zwischen den Webseiten:

$e = (u,v) \in E \Leftrightarrow$ Webseite u enthält einen Link auf Webseite v

Der Webspace bildet das Linkgeflecht der Webseiten im Internet ab. Der textuelle Inhalt der Webseiten wird dazu nicht berücksichtigt.

Adjazenzmatrix

Die **Adjazenzmatrix A** für einen gerichteten Graphen G = (V; E) ist eine (V×V)-Matrix. Sie ist wie folgt definiert:

$$A_{uv} = \begin{cases} 1 & \text{falls } (u, v) \in E \\ 0 & \text{sonst} \end{cases}$$

Eine Adjazenzmatrix kann auch Gewichte in Graphen berücksichtigen. Dazu wird an Stelle der Zahl 1 das Gewicht der jeweiligen Kante eingesetzt.

Authorities

Die Webseiten V können zwei verschiedenen Typen angehören:

Hubs (H) und **Authorities (A)**.

Die Zugehörigkeit ist graduell und nicht zwangsweise exklusiv.

Eine **Authority** ist eine Webseite, auf die viele Hubs verlinken. Das Maß a(v) für die Zugehörigkeit zum Typ Authority wird für eine Webseite v wie folgt berechnet:

$$a(v) := \sum_{u \in N_{in}(v)} h(u)$$

Dabei ist h(u) das Maß für die Zugehörigkeit der Webseite u zum Typ Hub. Der Wert a(v) wird auch als **Authority-Score** bezeichnet.

Hubs

Ein **Hub** ist eine Webseite, die Links auf viele Authorities enthält. Das Maß h(u) für die Zugehörigkeit zum Typ Hub berechnet sich für eine Webseite u wie folgt:

$$h(u) \coloneqq \sum_{v \in N_{out}(u)} a(v)$$

Dabei ist a(v) das Maß für die Zugehörigkeit der Webseite v zum Typ Authority. Der Wert h(u) wird auch als **Hub-Score** bezeichnet.

Hubs sind gewissermaßen populäre Linksammlungen (z.B. digg oder DMOZ).

Hubs & Authorities

Jede Webseite ist zu einem gewissen Grad **gleichzeitig Hub und Authority**. Meist wird eine Normalisierung der Zugehörigkeitsgrade gefordert:

$$\|a\| = 1 \Leftrightarrow \sqrt{\sum_{v \in V} a(v)^2} = 1 \Leftrightarrow \sum_{v \in V} a(v)^2 = 1$$

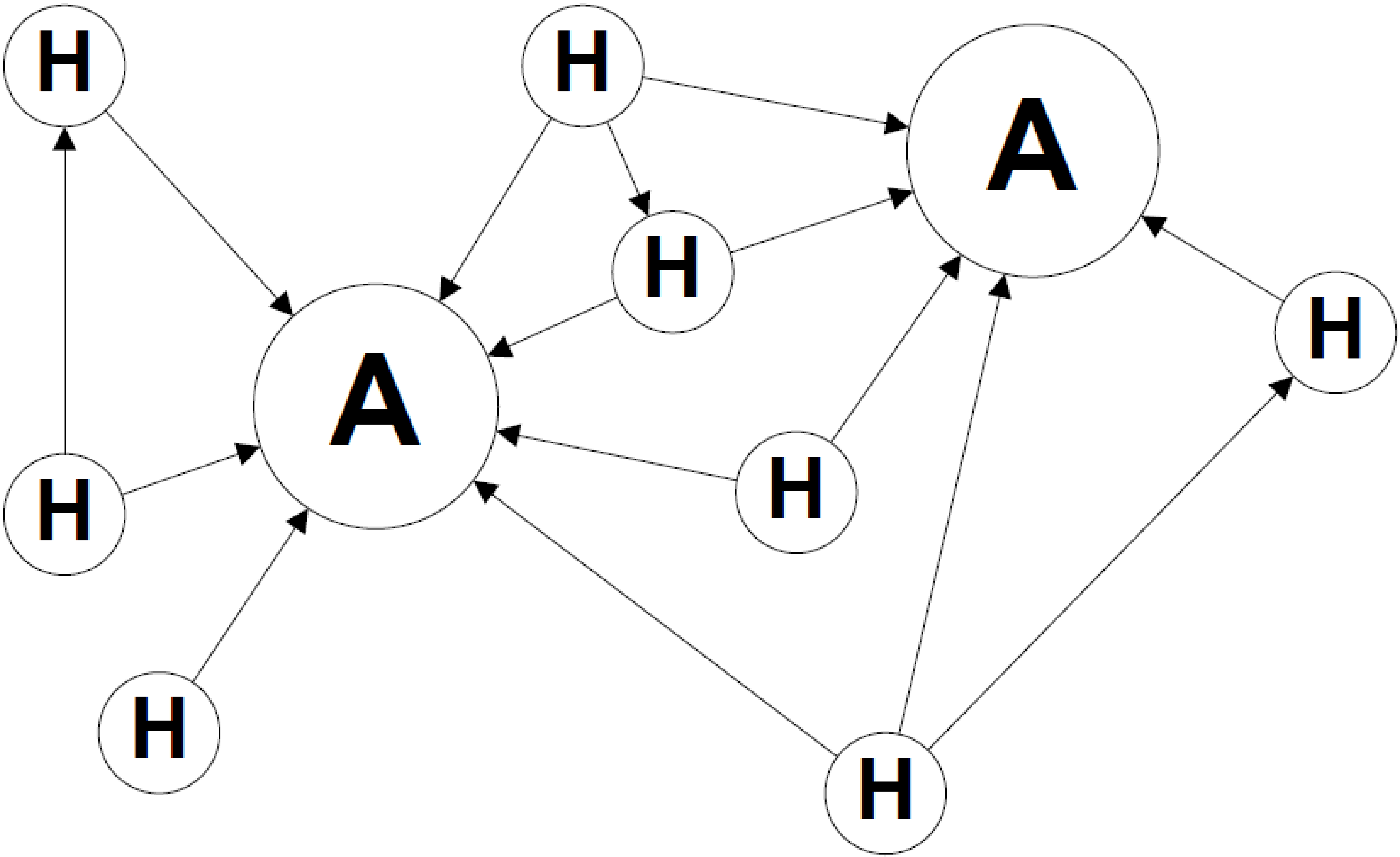
und

$$\|h\| = 1 \Leftrightarrow \sqrt{\sum_{u \in V} h(u)^2} = 1 \Leftrightarrow \sum_{u \in V} h(u)^2 = 1$$

Die Normalisierung eines n-dimensionalen Vektors x auf die Länge 1 erreichen wir, indem wir den Vektor durch seine Länge teilen:

$$norm(x) = \frac{1}{\|x\|} x = \frac{x}{\sqrt{\sum_{i=1}^{n} x_i^2}}$$

Hubs & Authorities



Hubs(H) und Authorities(A) für einen Webgraphen

HITS (Hyperlink-Induced Topic Search)

Sei W ein Webspace und G = (V,E) der zugeordnete gerichtete Graph mit |V|=N. Dann können die Authority-Scores a und Hub-Scores h folgendermaßen approximiert werden:

$$a_1 = \left(\sqrt{\frac{1}{N}}\right)_{i=1}^{N} \qquad h_1 = \left(\sqrt{\frac{1}{N}}\right)_{i=1}^{N}$$

$$a_{i+1} = \frac{1}{\|A^T h_i\|} \cdot A^T h_i \quad h_{i+1} = \frac{1}{\|A a_i\|} \cdot A a_i$$

$$a = \lim_{i \to \infty} a_i \qquad h = \lim_{i \to \infty} h_i$$

HITS-Algorithmus

HITS(Adjazenzmatrix A)

$i \leftarrow 0$

$a_1 \leftarrow \left(\sqrt{1/N}\right)_{i=1}^{N}$; $h_1 \leftarrow \left(\sqrt{1/N}\right)_{i=1}^{N}$

while $\delta > \epsilon$

do

    $i \leftarrow i + 1$

    $a_{i+1} \leftarrow A^T h_i$

    $\alpha \leftarrow 1/\|a_{i+1}\|$

    $a_{i+1} \leftarrow \alpha a_{i+1}$

    $h_{i+1} \leftarrow A a_i$

    $\eta \leftarrow 1/\|h_{i+1}\|$

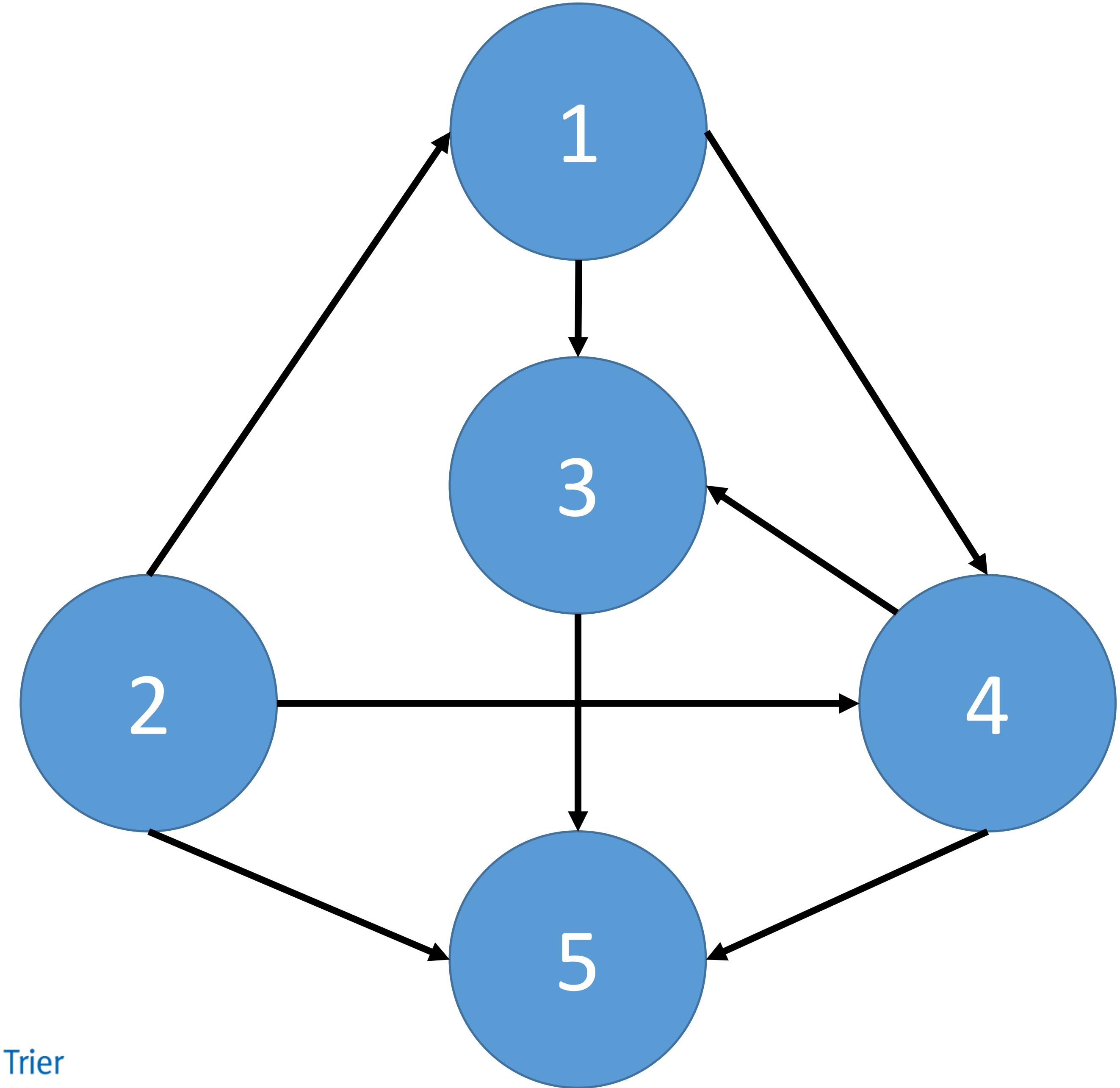
    $h_{i+1} \leftarrow \eta h_{i+1}$

    // δ ist monoton fallend

    $\delta \leftarrow \|h_{i+1} - h_i\| + \|a_{i+1} - a_i\|$

return (h_{i+1}, a_{i+1})

HITS: Beispiel

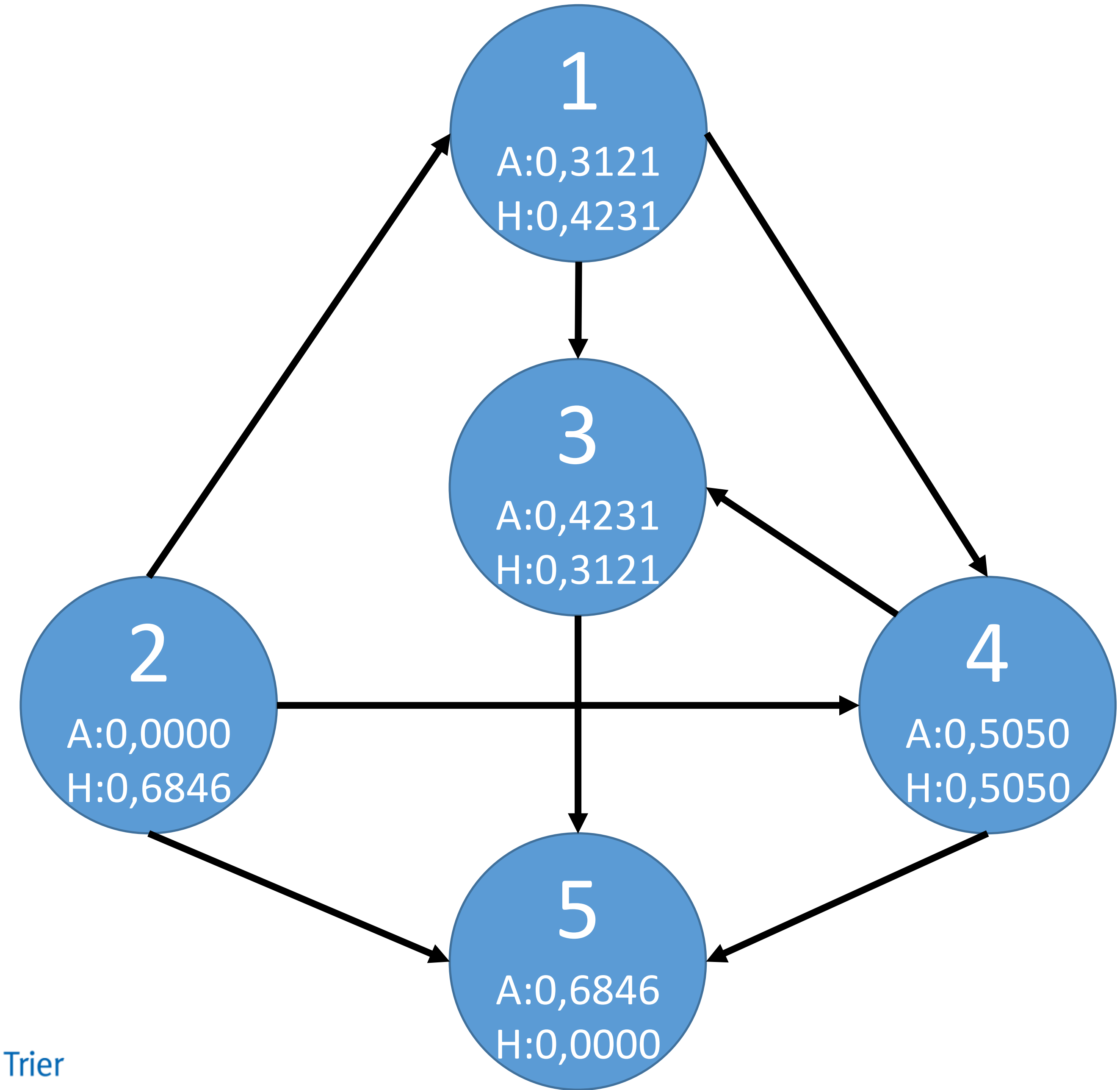


Ablauf des HITS-Algorithmus

Score	w_1	w_2	w_3	w_4	w_5	δ
a_1 h_1	0,4472 0,4472	0,4472 0,4472	0,4472 0,4472	0,4472 0,4472	0,4472 0,4472	–
a_2 h_2	0,2357 0,4714	0,0000 0,7071	0,4714 0,2357	0,4714 0,4714	0,7071 0,0000	1,93399993374627
a_3 h_3	0,3235 0,4313	0,0000 0,6470	0,4313 0,3235	0,5392 0,5392	0,6470 0,0000	0,51147794876831
a_4 h_4	0,2952 0,4429	0,0000 0,6889	0,4429 0,2952	0,4921 0,4921	0,6889 0,0000	0,25756387101341
a_5 h_5	0,3142 0,4263	0,0000 0,6732	0,4263 0,3142	0,5161 0,5161	0,6732 0,0000	0,15035043318893
a_6 h_6	0,3069 0,4297	0,0000 0,6855	0,4297 0,3069	0,5013 0,5013	0,6855 0,0000	0,07530034103623
a_7 h_7	0,3125 0,4244	0,0000 0,6810	0,4244 0,3125	0,5084 0,5084	0,6810 0,0000	0,04481279269550
a_8 h_8	0,3104 0,4253	0,0000 0,6847	0,4253 0,3104	0,5039 0,5039	0,6847 0,0000	0,02209746953076
...	...	...	...	...	...	...
a_{29} h_{29}	0,3121 0,4231	0,0000 0,6846	0,4231 0,3121	0,5050 0,5050	0,6846 0,0000	0,00000009075272

Annahmen: $\epsilon = 0{,}0000001$, Initialisierung von a und h mit $\left(\sqrt{0{,}2}\right)_{i=1}^{N}$

HITS: Beispiel

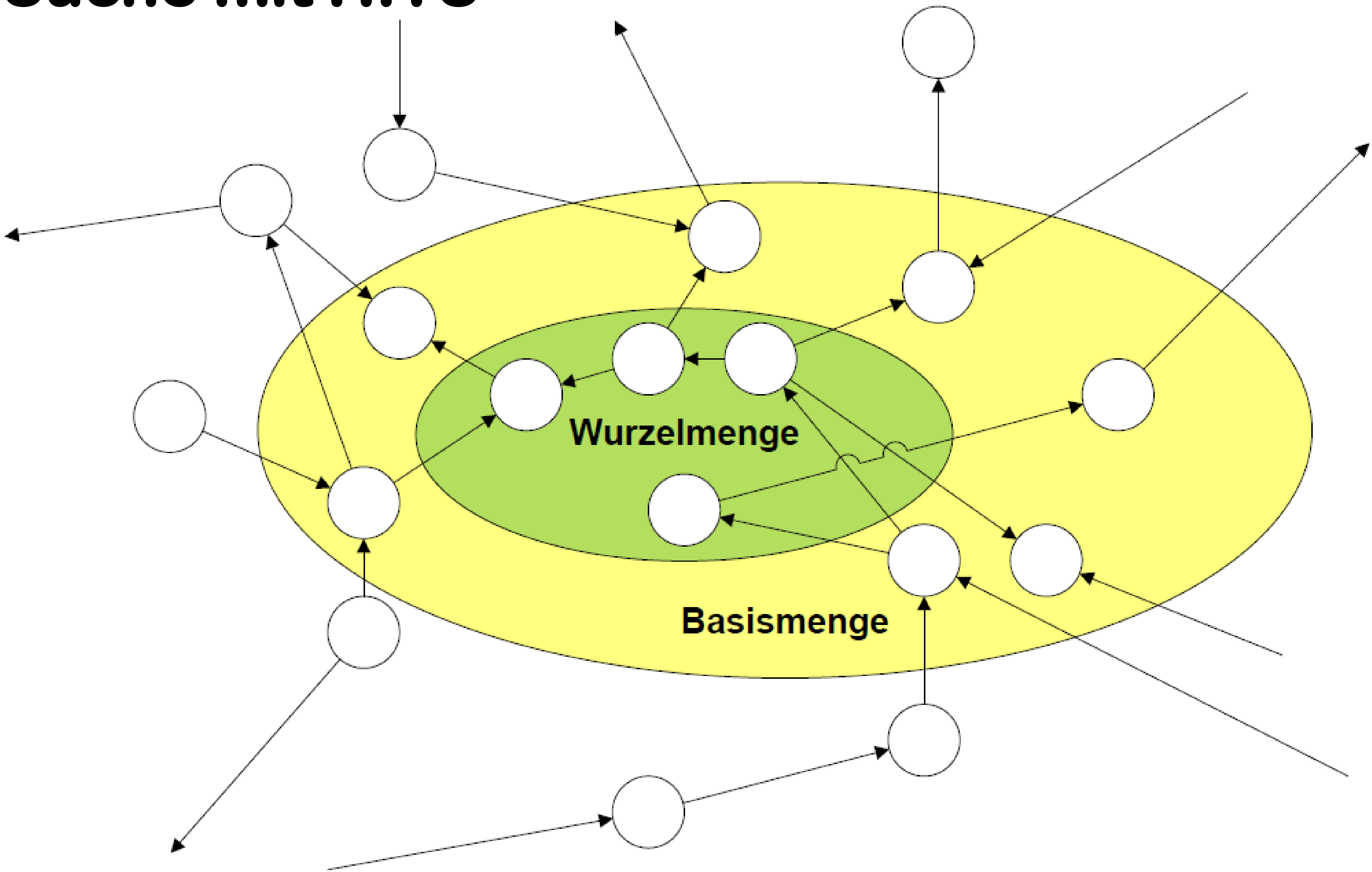


Suche mit HITS

Vorgehen bei der Beantwortung von Suchanfragen mit HITS-Ranking:

1. Bestimme anhand der Suchanfrage die **Wurzelmenge** aller Webseiten. Diese enthält alle Webseiten, welche die Suchbegriffe enthalten. (Empirisch: ca. 200 Webseiten sollten genügen.)
2. Bestimme anhand der Wurzelmenge die **Basismenge** aller Webseiten. Diese enthält die Wurzelmenge und sämtliche Webseiten, welche zu dieser verlinkt sind (Linkrichtung ist irrelevant).
3. Berechne Hub- und Authority-Scores der Webseiten in der Basismenge.
4. Gib die Webseiten in **absteigender Reihenfolge** entsprechend ihren **Authority-Scores** aus.

Suche mit HITS



Wurzelmenge und Basismenge zu einer Suchanfrage

Suche mit HITS

Beobachtungen zu Hubs und Authorities bzgl. einer Suchanfrage:

- Eine Webseite mit **gutem Authority-Score** könnte unter Umständen den Text der Suchanfrage gar nicht enthalten.
- Falls eine Webseite mit **gutem Hub-Score** den Suchtext enthält, sind häufig auch die **Authorities** gut, zu welchen die Webseite einen Link besitzt.
- Falls eine Webseite einen **guten Authority-Score** hat, sind häufig auch die **Hubs** gut, welche einen Link auf diese Webseite besitzen.

Vergleich PageRank – HITS

PageRank:

- Sehr große Matrix
- Hoher initialer Berechnungsaufwand
- Beantwortet Anfragen online sehr schnell
- Konvergenzgeschwindigkeit justierbar
- Weniger anfällig für Link-Spamming (betrachtet nur eingehende Links)
- Berechnet nur Authorities

Vergleich PageRank – HITS

HITS:

- Kleine Matrizen
- Kann Semantik von Anfragen berücksichtigen
- **Schwierig in Echtzeit**
- Anfällig für Link-Spamming (betrachtet nicht nur eingehende Links)
- Mindestens gleiche Ergebnisqualität wie PageRank
- **Tightly-Knit-Community-Effekt** (TKC-Effekt)
  HITS bevorzugt dicht vernetzte Gruppen und bewertet insbesondere kleine vollständige bipartite Graphen sehr hoch.
  Problematische Folgen:
  - Kleine Gruppen können den TKC-Effekt für Manipulation nutzen.
    Beispiel: Link-Spamming
  - Topic-Drift: Aufgrund des TKC-Effekts verschieben sich Anfrageergebnisse zu themenfremden, dichteren Communities. Beispiel: Genetics → Genetic Algorithms.
  - Polarisierte Communities verlinken sich nicht. Beispiel: Befürworter und Gegner von Trump