Denn $\det(A^T - \lambda\mathbb{I}) = \det(A^T - (\lambda\mathbb{I})^T) = \det((A - \lambda\mathbb{I})^T) = \det(A - \lambda\mathbb{I})$. Im letzten Schritt haben wir verwendet, dass wir wissen, dass eine Matrix dieselbe Determinante wie ihre transponierte Matrix hat.

Sehen wir uns ein weiteres Anwendungsbeispiel an.

> **Beispiel 14.15 Markov-Prozess**
> Ein Geschäft mit zwei Filialen verleiht tageweise Fahrräder. 60% der Fahrräder, die in der ersten Filiale ausgeliehen werden, werden dort auch zurückgegeben, der Rest in der anderen Filiale. 70% der Fahrräder, die in der zweiten Filiale ausgeliehen werden, werden auch dort zurückgegeben; der Rest wiederum in der anderen Filiale.
>
> Ist es möglich, die Fahrräder so auf beide Filialen zu verteilen, dass in jeder Filiale an jedem Morgen genau die gleiche Anzahl von Fahrrädern steht? Wenn man die Fahrräder irgendwie auf beide Filialen verteilt, was passiert dann im Laufe der Zeit?

Lösung zu 14.15 Bezeichnen wir die Anzahl der Fahrräder in den beiden Filialen am n-ten Tag mit $x_1(n)$ und $x_2(n)$, so gilt nach Voraussetzung für $\mathbf{x}(n) = (\mathbf{x}_1(n), \mathbf{x}_2(n))$, dass

$$\mathbf{x}(n+1) = A\mathbf{x}(n), \qquad \text{mit} \quad A = \begin{pmatrix} 0.6 & 0.3 \\ 0.4 & 0.7 \end{pmatrix}.$$

Für die gewünschte Verteilung $\mathbf{x}$, bei der die morgendliche Anzahl der Fahrräder in den beiden Filialen jeden Tag gleich ist, muss

$$\mathbf{x} = A\mathbf{x}$$

gelten, sie muss also ein Eigenvektor zum Eigenwert eins sein. Setzen wir das charakteristische Polynom gleich null,

$$\lambda^2 - 1.3\lambda + 0.3 = 0,$$

so folgen die Eigenwerte $\lambda_1 = 1$ und $\lambda_2 = 0.3$. Die zugehörigen Eigenvektoren lauten

$$\mathbf{u}_1 = k_1 \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \qquad \mathbf{u}_2 = k_2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad \text{mit } k_1, k_2 \in \mathbb{R}.$$

Die gesuchte Verteilung ist der Eigenvektor zum Eigenwert $\lambda_1 = 1$,

$$\mathbf{x} = \frac{1}{7}\begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.57 \end{pmatrix},$$

(die Länge des Eigenvektors wurde hier so gewählt, dass $x_1 + x_2 = 1$ ist). In der ersten Filiale sollten an einem Morgen also 43% und in der zweiten 57% der Fahrräder sein, dann wird sich an dieser Verteilung auch in Zukunft nichts mehr ändern.

Wie sieht es nun mit dem Verhalten im Lauf der Zeit aus? Ist irgendeine Verteilung $\mathbf{x}(0)$ am Anfang gegeben, so ist die Verteilung nach n Tagen

$$\mathbf{x}(n) = A^n \mathbf{x}(0).$$

Das ist zwar eine nette Formel, die man mit dem Computer für jedes n auswerten kann, aber was im Lauf der Zeit passiert, ist daraus nicht ablesbar! Gehen wir zur Basis aus Eigenvektoren über,

$$\mathbf{x}(0) = U\mathbf{y} = y_1\mathbf{u}_1 + y_2\mathbf{u}_2,$$

so erhalten wir nach n Tagen

$$\mathbf{x}(n) = A^n\mathbf{x}(0) = y_1A^n\mathbf{u}_1 + y_2A^n\mathbf{u}_2 = y_1\lambda_1^n\mathbf{u}_1 + y_2\lambda_2^n\mathbf{u}_2 = y_1\mathbf{u}_1 + y_2(0.3)^n\mathbf{u}_2.$$

Die Komponente in $\mathbf{u}_2$-Richtung nimmt also exponentiell ab, daher konvergiert die Verteilung gegen die Gleichgewichtsverteilung $y_1\mathbf{u}_1$. Mit anderen Worten, vollkommen unabhängig davon, mit welcher Verteilung man startet, wird sich im Lauf der Zeit diese Gleichgewichtsverteilung einstellen. ■

Allgemein zeigt letztes Beispiel unter anderem, dass A^k $(k \in \mathbb{N})$ für eine diagonalisierbare Matrix leicht mittels

$$A^k = U \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix} U^{-1}$$

berechnet werden kann.

Denn für $A = U^{-1}BU$ gilt: $A^k = A \cdot A \cdots A = (U^{-1}BU)(U^{-1}BU)\cdots(U^{-1}BU) = U^{-1}B(UU^{-1})B \cdots (UU^{-1})BU = U^{-1}B^kU$.

Hier erweist es sich auch für reelle Matrizen als sinnvoll, komplexe Eigenwerte zuzulassen! Denn auch wenn die Bestandteile in $U\mathrm{diag}(\lambda_1^k, \dots, \lambda_n^k)U^{-1}$ komplex sind: Das Endergebnis ist A^k und damit reell.

Das Fahrradproblem ist ein Beispiel für einen **Markov-Prozess** (benannt nach dem russischen Mathematiker Andrej Andrejewitsch Markov, 1856–1922). Das ist ein stochastischer Prozess, bei dem die Wahrscheinlichkeit, einen bestimmten Zustand zu erreichen, *nur* vom vorhergehenden Zustand abhängt. Ein Markov-Prozess kann mithilfe einer Matrix beschrieben werden: Wenn A die Matrix ist, die den Übergang von einem Zustand in den darauf folgenden beschreibt, und $\mathbf{x}$ der Anfangszustand, dann ist $\mathbf{y} = A^n\mathbf{x}$ der Zustand nach n Schritten.

Die charakteristische Eigenschaft der Matrix A ist dabei, dass alle Koeffizienten nichtnegativ und die Spaltensummen immer gleich eins sind. Eine Matrix mit dieser Eigenschaft wird auch als **Markov-Matrix** oder **stochastische Matrix** bezeichnet.

Satz 14.16 Eine Markov-Matrix hat immer den Eigenwert eins und es gibt dazu immer einen Eigenvektor, dessen Komponenten alle nichtnegativ sind.

Dass A immer den Eigenwert eins hat, ist leicht zu sehen: Die transponierte Matrix A^T hat Zeilensummen eins, und deshalb ist $A^T\mathbf{e} = \mathbf{e}$, das heißt, $\mathbf{e} = (1, 1, \dots, 1)$ ist ein Eigenvektor zum Eigenwert eins. Damit hat auch A den Eigenwert eins, denn A und A^T haben die gleichen Eigenwerte.

Eine beliebige Anfangsverteilung muss aber nicht immer gegen einen Gleichgewichtszustand konvergieren, sie könnte auch hin und her springen, wie zum Beispiel bei

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

(Diese Matrix hat die Eigenwerte 1 und −1.) Wann ist die Konvergenz gegen einen Gleichgewichtszustand gegeben?

Satz 14.17 Sei A eine Markov-Matrix. Dann sind alle Eigenwerte vom Betrag kleiner gleich eins. Gibt es außer eins keinen Eigenwert mit Betrag gleich eins, so konvergiert für einen beliebigen Anfangszustand $\mathbf{x}(0)$ die Folge von Vektoren $\mathbf{x}(n) = A^n\mathbf{x}(0)$ gegen einen Gleichgewichtszustand (der vom Anfangszustand abhängen kann).

Diese Situation war zum Beispiel beim Fahrradproblem gegeben.

Man kann zeigen, dass die Bedingung aus Satz 14.17 zum Beispiel erfüllt ist, falls alle Diagonalelemente der Matrix A positiv sind. Sind sogar alle Koeffizienten von A positiv, so ist der Gleichgewichtszustand eindeutig.

Übrigens, auch lineare Rekursionen lassen sich als Eigenwertproblem behandeln, wenn man sie etwas umformuliert. Zum Beispiel können wir die Fibonacci-Rekursion

$$x_{n+1} = x_n + x_{n-1}$$

als

$$\mathbf{x}(n+1) = A\mathbf{x}(n)$$

schreiben, wenn wir

$$\mathbf{x}(n) = \begin{pmatrix} x_n \\ x_{n-1} \end{pmatrix} \qquad \text{und} \qquad A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

setzen. (Natürlich ist A im Allgemeinen keine Markov-Matrix mehr.)

14.2.1 Anwendung: Bewertung von Webseiten mit *PageRank*

Markov-Prozesse werden auch bei der Suchmaschine Google verwendet. Die Idee dahinter wollen wir uns etwas genauer ansehen.

Wir haben eine Anzahl von n Seiten, die miteinander verlinkt sind, gegeben. Die Links werden durch eine Matrix L_{ij} beschrieben:

$$L_{ij} = \begin{cases} 1, & \text{falls ein Link von Seite } i \text{ zur Seite } j \text{ besteht} \\ 0, & \text{sonst} \end{cases}$$

Üblicherweise zählt man Links einer Seite auf sich selbst nicht und setzt L_{ii} in jedem Fall null.

Aufgabe einer guten Suchmaschine ist, nicht nur Seiten, die ein bestimmtes Stichwort enthalten, zu finden, sondern auch die Treffer zu sortieren. Dazu ist es notwendig, alle Seiten zu bewerten und jeder Seite ein Gewicht x_j zuzuordnen. Für ein sinnvolles Gewicht kann man jeden Link auf eine Seite als „Stimme“ für diese Seite ansehen. Im einfachsten Fall zählt man also die Anzahl der Seiten, die auf die i-te Seite verlinken, und definiert

$$x_i = \sum_{j=1}^{n} L_{ji}$$

als deren Gewicht. Ein Nachteil dabei ist, dass die Stimme einer Seite mit vielen Links genauso viel zählt, wie die Stimme einer Seite mit wenigen ausgesuchten Links. Deshalb verfeinern wir unsere Gewichts-Definition und geben jeder Seite nur eine Stimme, die sie gleichmäßig auf alle Seiten, auf die sie verlinkt, „verteilt“:

$$x_i = \sum_{j=1}^{n} \frac{1}{n_j} L_{ji}, \qquad \text{mit} \qquad n_j = \sum_{i=1}^{n} L_{ji},$$

wobei n_j also die Anzahl der Links auf der j-ten Seite ist.

Sollte eine Seite nicht von ihrem Stimmrecht Gebrauch machen, sollte also $n_j = 0$ sein, so ist der Term $\frac{1}{n_j} L_{ji}$ durch 0 zu ersetzen.

Das ist schon besser, aber immer noch nicht optimal. Der Betreiber einer Seite könnte einfach eine große Anzahl von weiteren Seiten erstellen, deren einziger Zweck es ist, auf seine eigentliche Seite zu verlinken, um deren Bewertung zu erhöhen. Insbesondere würden große Websites mit vielen untereinander verlinkten Seiten automatisch besser bewertet als kleinere Websites. Wir müssen also nochmals nachbessern, indem wir nicht jeder Seite genau eine Stimme geben, sondern genau so viel Stimmrecht, wie es ihrem *Gewicht* entspricht

$$x_i = \sum_{j=1}^{n} \frac{1}{n_j} L_{ji} x_j.$$

Hoppla, werden Sie sich jetzt vielleicht denken, da drehen wir uns aber im Kreis: Auf der rechten Seite kommen wieder die Gewichte x_j vor, die wir ja gerade ausrechnen möchten! Stimmt, wir haben eben eine Gleichung

$$\mathbf{x} = A\mathbf{x} \quad \text{mit} \quad A_{ij} = \frac{1}{n_j} L_{ji}$$

für die gesuchten Gewichte $\mathbf{x}$ bekommen. Nach Konstruktion ist A eine Markov-Matrix und die gesuchten Gewichte ergeben einen Gleichgewichtszustand.

Im Fall $n_j = 0$ (Seite ohne Links) haben wir wieder das Problem, dass die j-te Spalte von A gleich null ist, obwohl die Spaltensummen einer Markov-Matrix gleich eins sein müssen. Diese Seite hat aber keinen Einfluss auf die Gewichte der anderen Seiten und wir können daher diese Seite einfach entfernen. Ihr Gewicht kann dann später leicht aus den Gewichten der übrigen Seiten berechnet werden.

Um die gesuchten Gewichte zu erhalten, müssen wir also einen Eigenvektor von A zum Eigenwert eins finden, also das lineare Gleichungssystem $(A - \mathbb{I})\mathbf{x} = 0$ lösen. Theoretisch ist das kein Problem, aber bei der Anzahl der Seiten im Internet sind mit dieser Aufgabe auch die derzeit schnellsten Computer überfordert. Was also tun? Muss unsere schöne Idee mangels praktischer Durchführbarkeit in den Papierkorb wandern? Nein! Bei einem Markov-Prozess kann man den Gleichgewichtszustand ja näherungsweise durch die Iteration

$$\mathbf{x}(k+1) = A\mathbf{x}(k)$$

eines Anfangszustands $\mathbf{x}(0)$ erhalten. Diese Iteration kann (vergleichsweise) schnell berechnet werden, da die meisten Koeffizienten von A ja null sind (denn eine Seite verlinkt nur auf einen Bruchteil aller anderen Seiten im Internet). Man kann dadurch zu einer, für unsere Zwecke vollkommen ausreichenden, Näherung für den Gleichgewichtszustand gelangen.

Man kann sich die Iteration auch wie eine Anzahl von Zufallssurfern vorstellen. Der Vektor $\mathbf{x}(k)$ gibt an, wie viele Surfer sich im k-ten Schritt auf jeder einzelnen Seite befinden. In jedem Schritt sucht sich jeder Surfer zufällig einen Link auf seiner aktuellen Seite aus und wechselt auf diese nächste Seite. Im Lauf der Zeit stellt sich dabei eine Gleichgewichtsverteilung der Surfer ein. Der Prozentsatz der Surfer pro Seite entspricht dann dem Gewicht der Seite.

Wir sind somit fast am Ziel angelangt, nur eine einzige kleine Hürde ist noch zu nehmen: Wir haben ja im letzten Abschnitt gelernt, dass die Iteration eines Markov-Prozesses nicht immer konvergiert. Wenn zwei Seiten nur auf die jeweils andere verlinken, so springt die Iteration immer hin und her.

Ein Zufallssurfer, der in diese Falle tappt, wäre also gefangen und auf ewig dazu verdammt, zwischen diesen beiden Seiten hin und her zu wechseln.

Deshalb führen wir noch einen Dämpfungsfaktor $\alpha \in (0,1)$ ein und legen fest, dass nur der Anteil α jedes Gewichtes über die Bewertung durch andere Seiten, und der verbleibende Anteil $1-\alpha$ des Gewichtes fix (d.h. unabhängig von der Bewertung durch andere Seiten) erhalten wird.

Im Bild der Zufallssurfer bedeutet das, dass nur der Bruchteil α aller Zufallssurfer aus den Links der aktuellen Seite wählt, und der Rest $1-\alpha$ sich zufällig irgendeine neue Seite aussucht.

Wir haben festgelegt, dass jede Seite im Durchschnitt Gewicht eins hat, also

$$\sum_{j=1}^{n} x_j = n.$$

Eine Seite mit Gewicht größer eins ist also überdurchschnittlich gut, eine Seite mit Gewicht kleiner eins unterdurchschnittlich gut bewertet. Mit dem Dämpfungsfaktor α erhalten wir nun folgendes modifizierte Gleichungssystem für die Gewichte:

$$\mathbf{x} = (1-\alpha)\mathbf{e} + \alpha A\mathbf{x},$$

wobei $\mathbf{e} = (1, 1, \ldots, 1)$. In der Praxis wird ein Wert um $\alpha = 0.85$ verwendet. Das bedeutet, dass jede Seite ein Grundgewicht von 0.15 erhält, den Rest ihres Gewichtes erhält sie über die Bewertung durch andere Seiten.

Die Lösung dieses Gleichungssystems ist

$$\mathbf{x} = (1-\alpha)(\mathbb{I} - \alpha A)^{-1}\mathbf{e}.$$

Da ein exakte Lösung, wie schon erwähnt, zu aufwändig ist, bestimmen wir sie mittels Iteration:

$$\mathbf{x}(k+1) = (1-\alpha)\mathbf{e} + \alpha A\mathbf{x}(k).$$

Zwei Fragen sind noch zu klären. Erstens müssen wir sicherstellen, dass die Matrix $\mathbb{I} - \alpha A$ invertierbar ist (denn nur dann hat das Gleichungssystem eine eindeutige Lösung): Die Eigenwerte von A sind vom Betrag kleiner gleich eins. Also sind die Eigenwerte von αA kleiner gleich $\alpha < 1$ (denn die Eigenwerte von αA erhält man, wenn man die Eigenwerte von A mit α multipliziert). Somit ist eins kein Eigenwert von αA und damit ist $\mathbb{I} - \alpha A$ invertierbar (wäre eins ein Eigenwert von αA, so wäre $\det(\mathbb{I} - \alpha A) = 0$).

Zweitens müssen wir überlegen, dass die Iteration konvergiert: Berechnen wir dazu den Zustand nach k Schritten,

$$\mathbf{x}(k) = (1-\alpha)\sum_{j=0}^{k-1} \alpha^j A^j \mathbf{e} + \alpha^k A^k \mathbf{x}(0).$$

Für die Markov-Matrix A bleiben die Vektoren $A^k\mathbf{x}(0)$ beschränkt (sie konvergieren gegen einen Gleichgewichtszustand oder springen hin und her) und der Faktor α^k bewirkt, dass $\alpha^k \|A^k \mathbf{x}(0)\| \to 0$ konvergiert. Das Gleiche gilt für die Vektoren $A^j\mathbf{e}$ und aus der Konvergenz der geometrischen Reihe $\sum_{j=0}^{\infty} \alpha^j$ folgt (mit dem Majorantenkriterium) die Konvergenz der Reihe $\sum_{j=0}^{\infty} \alpha^j A^j \mathbf{e}$. Somit konvergiert die Iteration.

Dieser Algorithmus zur Webseitenbewertung bildet das Herzstück von Google und ist als PageRank (http://www-db.stanford.edu/˜Ebackrub/google.html) bekannt.

14.3 Eigenwerte symmetrischer Matrizen

Im letzten Abschnitt haben wir gesehen, dass Eigenwerte im Allgemeinen auch komplex sein können, und dass es unter Umständen nicht genügend linear unabhängige Eigenvektoren gibt, um eine Matrix zu diagonalisieren. In der Praxis hat man es jedoch oft mit *symmetrischen Matrizen* zu tun, also Matrizen mit $A = A^T$ (bzw. im Fall komplexer Matrizen $A = A^*$), die folgende angenehme Eigenschaften haben:

Satz 14.18 Die Eigenwerte einer symmetrischen Matrix sind reell und die Eigenvektoren zu verschiedenen Eigenwerten sind orthogonal.

Zuerst zur Eigenschaft, dass die Eigenwerte reell sind: Wir gehen von der Beziehung $\langle A\mathbf{x}, \mathbf{y}\rangle = \langle \mathbf{x}, A\mathbf{y}\rangle$ aus, die für symmetrische Matrizen und komplexes Skalarprodukt gilt (siehe Satz 13.3): Wenn $A\mathbf{u} = \lambda\mathbf{u}$ gilt, so erhalten wir $\lambda\langle \mathbf{u}, \mathbf{u}\rangle = \langle \mathbf{u}, \lambda\mathbf{u}\rangle = \langle \mathbf{u}, A\mathbf{u}\rangle = \langle A\mathbf{u}, \mathbf{u}\rangle = \langle \lambda\mathbf{u}, \mathbf{u}\rangle = \overline{\lambda}\langle \mathbf{u}, \mathbf{u}\rangle$. Da für $\mathbf{u} \neq \mathbf{0}$ (das gilt ja per Definition für Eigenvektoren) $\langle \mathbf{u}, \mathbf{u}\rangle = \|\mathbf{u}\|^2 > 0$ ist, können wir auf beiden Seiten kürzen und erhalten $\lambda = \overline{\lambda}$. Also sind die Eigenvektoren reell.

Um die zweite Eigenschaft zu sehen, gehen wir analog vor: Aus $A\mathbf{u}_1 = \lambda_1\mathbf{u}_1$ und $A\mathbf{u}_2 = \lambda_2\mathbf{u}_2$ folgt $\lambda_1\langle \mathbf{u}_1, \mathbf{u}_2\rangle = \langle \lambda_1\mathbf{u}_1, \mathbf{u}_2\rangle = \langle A\mathbf{u}_1, \mathbf{u}_2\rangle = \langle \mathbf{u}_1, A\mathbf{u}_2\rangle = \langle \mathbf{u}_1, \lambda_2\mathbf{u}_2\rangle = \lambda_2\langle \mathbf{u}_1, \mathbf{u}_2\rangle$. Es gilt also $(\lambda_1 - \lambda_2)\langle \mathbf{u}_1, \mathbf{u}_2\rangle = 0$ und ist $\lambda_1 \neq \lambda_2$, so muss $\langle \mathbf{u}_1, \mathbf{u}_2\rangle = 0$ gelten.

Es gilt sogar noch mehr, denn man kann folgendes zeigen:

Satz 14.19 Jede symmetrische Matrix ist diagonalisierbar und die Eigenvektoren können so gewählt werden, dass sie eine Orthonormalbasis bilden.

Wie erhält man nun eine Orthonormalbasis aus Eigenvektoren? Nach Satz 14.18 sind Eigenvektoren zu verschiedenen Eigenwerten automatisch orthogonal; es genügt also, sie zu normieren. Linear unabhängige Eigenvektoren zum gleichen Eigenwert sind zwar nicht automatisch orthogonal, wir können sie aber jederzeit mit dem Gram-Schmidt-Verfahren orthonormalisieren (denn Linearkombinationen von Eigenvektoren zum gleichen Eigenwert sind ja wieder Eigenvektoren zu diesem Eigenwert).

Beispiel 14.20 Hauptachsentransformation
Gegeben ist die quadratische Kurve

$$\langle \mathbf{x}, A\mathbf{x}\rangle = 4, \qquad \text{mit} \qquad A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \text{ und } \mathbf{x} \in \mathbb{R}^2.$$