

Eigenwerte, Eigenvektoren

Stochastische Prozesse Haftendorn 2011

Eigenwerte (Zur Erzeugung der stochastischen Matrizen siehe 2. Problem)

Es gibt eingebaute Befehle dafür `eigVl(matrix)` `eigVc(matrix)`. Weiter gibt es eine Bibliothek `linalgcas.tns`, sie enthält `linalgcas:eigenval(matrix)` und `linalgcas:eigenvect(matrix)`.

`aa := [0.73 0.13 0.14; 0.43 0.43 0.14; 0.15 0.15 0.7]` Die Matrix ist mit `toolstomat(3)` erzeugt, dann aber mit copy und paste dem `aa` direkt zugewiesen, damit bei anderer Auswertung keine andere Matrix erzeugt wird.

`eigVl(aa) * [1, 0, 3, 56]` Der eingebaute Befehl versagt aber bei komplexen Eigenwerten, wenn er im real-Modus ist. In der Stochastik interessiert sowieso nur der Eigenwert 1, mit

`ee := identity(3) * [1 0 0; 0 1 0; 0 0 1]`

und an `det(aa - ee) = 0` kann man sehen, dass die Matrix `aa` den Eigenwert 1 wirklich hat.

Eine Alternative ist aus dem Bibliotheksprogramm `linalgcas.tns` zu nehmen `linalgcas:eigenval(aa) * {0, 3, 0, 56, 1}`, aber man weiß es ja schon, es gibt den Eigenwert 1. Übrigens kann man mit den anderen Eigenwerten keine stochastischen Prozesse beschreiben, weil `v` auf z.B. $0.3 \cdot v$ abgebildet würde, das ja kein stochastischer Vektor ein, wenn `v` einer ist.

1.1

Eigenwerte, Eigenvektoren

Weitere Beispiele, Matrizen erzeugt mit toolstomat(3)

`a1 := [0.32 0.32 0.36; 0.86 0.05 0.09; 0.93 0.06 0.01]` `[0.32 0.32 0.36; 0.86 0.05 0.09; 0.93 0.06 0.01]` `det(a1 - ee) = 0` also 1 ist Eigenvektor

Bestimmung des zugehörigen Eigenvektors

`to1 := solve(augment(mat*list([x y z]:a1*[x y z]),{x,y,z}),{x,y,z})`
`* x=0.568692 and y=0.205802 and z=0.225506`

Der Eigenvektor lässt sich auch aus der hohen Potenz ablesen.

`stabil1 := a1^30 * [0.568692 0.205802 0.225506; 0.568692 0.205802 0.225506; 0.568692 0.205802 0.225506]`

`eva1 := stabil1[1]` `[0.568692 0.205802 0.225506]` Dies ist der stabile Vektor

Probe `eva1 * a1 = [0.568692 0.205802 0.225506]`

`[333 333 334] * a1^30 = [568.692 205.802 225.506]`

Von 1000 Leuten mit zunächst gleichverteilten Merkmalen haben auf lange Sicht 567 Merkmal 1, 206 Merkmal 2 und 225 haben Merkmal 3.

1.3

Eigenwerte, Eigenvektoren

Die in der Bibliothek `linalgcas.tns` vorhandene Funktion

`linalgcas:eigenvect(aa, 1)` liefert hier erstmal Unsinn, da sich das Konzept auf Spaltenvektoren bezieht. Darum wird `aa` transponiert, das Zeichen `*` ist bei den Sonderzeichen.

`linalgcas:eigenvect(aa', 1)` `[1; 0.4; 0.653333]`

`nor := sum(mat*list(evt))` Fehler: Ungültiger Datentyp `[0.4; 0.653333]` Fehler: Ungültiger Datentyp

Also mit dem Trick schafft man es. Ebenso mit der im TI Nspire eingebauten Funktion.

`eigVc(aa') * [0.79384 -0.731792 0.707107; -0.317536 0.052271 -0.707107; -0.518642 0.679521 -1.44254e15]` `eve := [0.79384 0.79384; -0.317536 0.317536; -0.518642 0.518642]`

`noe := sum(mat*list(eve))` `1.63002` `[0.487013 0.194805 0.318182]`

Die erste Spalte musste man nehmen, da eigV die 1 zuerst genannt hatte (s.o.). Außerdem können die anderen wegen der Vorzeichen nicht auf stochastische Vektoren umgeschrieben werden. Aber wie gesagt, der Aufwand lohnt sich nicht, aber schön wenn alles passt.

1.5

Stochastische Matrizen, einfach

Beispiel für stochastische Matrix `stomat(n)` ..

`aa := stma(3) * [0.67 0.27 0.06; 0.02 0.02 0.96; 0.78 0.11 0.11]` erzeugt stochastische nxn Zufallsmatrix

Wenn einem die nicht gefällt, kann man sie ja modifizieren. Siehe auch "Stoch- Matrizen"

Dabei sind die Zeilensummen 1 `start := [10 30 40]` `[10 30 40]`

`start-aa * [38.5 7.7 33.8]` `aa^20 * [0.575742 0.18543 0.238828; 0.575742 0.18543 0.238828; 0.575742 0.18543 0.238828]`

Die `end := start-aa^20 * [46.0593 14.8344 19.1063]`

$\sum_{i=1}^3 (start[i, j]) \cdot 80 = \sum_{i=1}^3 (end[i, j]) \cdot 80$.

`aa^80 * [0.575742 0.18543 0.238828; 0.575742 0.18543 0.238828; 0.575742 0.18543 0.238828]` gibt die langfristige Verteilung der 80 Leute an.

2.2

Eigenwerte, Eigenvektoren

Eigenvektoren ..

Am einfachsten bestimmt man den stabilen Vektor `v`, der `v-aa*v` erfüllt, aus einer hohen Matrixpotenz `stabil-aa^30 * [0.487013 0.194805 0.318182; 0.487013 0.194805 0.318182; 0.487013 0.194805 0.318182]`

Wenn die Zeilen sind gleich sind, ist der Exponent hoch genug gewählt. Jede Zeile ist der gesuchte Eigenvektor.

Probe: `evp := stabil[1] * [0.487013 0.194805 0.318182]`

`evp-aa^30 * [0.487013 0.194805 0.318182]`

Theoretisch fundierte Bestimmung: Das übliche Eigenvektor-Konzept arbeitet mit Spaltenvektoren. Darum ist es hier einfacher, den einzigen Eigenvektor, der interessiert, nämlich den zum Eigenwert 1 direkt aus einer Gleichung zu bestimmen. Die Gleichung `v-aa*v` muss also erfüllt werden und `v` muss ein stochastischer Vektor sein, also Spaltensumme 1 haben. Dazu müssen wir mit `mat*list` aus der Vektorgleichung ein Gleichungssystem machen. Dem wir die Normierungsgleichung noch hinzufügen.

`too := solve(augment(mat*list([x y z]:aa*[x y z]),{x+y+z=1}),{x,y,z})`
`* x=0.487013 and y=0.194805 and z=0.318182`

Dies sind die Komponenten des stabilen Vektors, des Eigenvektors.

1.2

Eigenwerte, Eigenvektoren

Weitere Beispiele, Matrizen erzeugt mit toolstomat(3)

`a2 := [0.12 0.12 0.76; 0.11 0.11 0.78; 0.4 0.4 0.2]` `det(a2 - ee) = 0` also 1 ist Eigenwert

Bestimmung des zugehörigen Eigenvektors

`to2 := solve(augment(mat*list([x y z]:a2*[x y z]),{x,y,z}),{x,y,z})`
`* x=0.254777 and y=0.254777 and z=0.490446`

Der Eigenvektor lässt sich auch aus der hohen Potenz ablesen.

`stabil2 := a2^30 * [0.254777 0.254777 0.490446; 0.254777 0.254777 0.490446; 0.254777 0.254777 0.490446]`

`eva2 := stabil2[1]` `[0.254777 0.254777 0.490446]` Dies ist der stabile Vektor

Probe `eva2 * a2 = [0.254777 0.254777 0.490446]`

`[333 333 334] * a2^30 = [254.777 254.777 490.446]`

Von 1000 Leuten mit zunächst gleichverteilten Merkmalen haben auf lange Sicht 255 Merkmal 1, 255 Merkmal 2 und 490 haben Merkmal 3.

1.4

Stochastische Matrizen, einfach

Stochastische Matrizen (einfache Version) Haftendorn 2011

Im Umgang mit stochastischen Prozessen, wie Markov-Ketten, gibt es Zustandsvektoren und stochastische Matrizen. In der Fachliteratur sind die ersteren Zeilenvektoren und die Matrizen haben Zeilensumme 1. In Schulbüchern wird es z.T. anders herum gemacht, um keinen Wechsel zu den geometrisch gedeuteten Vektoren der linearen Algebra zu haben. Das hat didaktischen Wert. Hier folge ich der Fachliteratur, Zustandsvektor z.B. `v := [0.1 0.3 0.6]` `[0.1 0.3 0.6]`

Unten ist die Programmierung für die Funktion `stma(n)` angegeben.

`sa := stma(3) * [0.1 0.1 0.8; 0.95 0.03 0.02; 0.76 0.06 0.18]` und nun `v * sa = [0.751 0.055 0.194]` Vektor links!!!!

Stochastische Matrizen haben Zeilensumme 1

Im Bibliotheksprogramm `tool.tns` ist der entsprechende Befehl `toolstomat(3) * [0.42 0.42 0.16; 0.59 0.4 0.01; 1. 0. 0.]`

den braucht man, wenn man im mehreren Problemen einer oder mehrerer Dateien stochastische Zufalls-Matrizen braucht.

Allerdings haben die Komponentennicht den Erwartungswert `1/n` wie es eigentlich sein müsste. Dieser Mangel wird von der Programmierung in der Datei `StochastischeMatrizen.tns` behoben.

2.1

Stochastische Matrizen, einfach

`stma` 2/15

Define LibPub `stma(n)`

Func

© `stma(n)` gibt eine quadr. stochastische Matrixdim `nxn` aus.

Local `i,j,ma,lima,h,z`: `ma:=newMat(n,n)`; `lima:={}`

For `i,1,n`

`z:=randInt(0,100)`; `h:={z}`

For `j,2,n-1`

While `sum(h)+z>100`

`z:=randInt(0,100)`

EndWhile

`h:=augment(h,{z})`

EndFor

`z:=100-sum(h)`; `h:=augment(h,{z})`; `h:=toolperm(h)`

`lima:=augment(lima,h)`

EndFor

`lima:=lima-0.01`

Return `list*mat(lima,n)`

EndFunc

Die stochastische Matrix wird zeilenweise aus `n` stochastischen Vektoren erzeugt: Es wird in der Einheit % von 0 bis 100 ein erster Wert zufällig erzeugt und in die Liste geschrieben. Ein weiterer Wert kann den Rest bis 100 ausschöpfen u.s.w..

Damit nicht immer die großen Werte vorn stehen, wird die Liste `h` noch zufällig permutiert. Damit ist ein stochastischer Vektor erzeugt.

Am Ende werden die Prozente noch richtig als Dezimalzahlen zwischen 0 und 1 angegeben.

2.3