Einführung in Sage - Einheit 4

Matrizen, Vektorräume, Funktionen

Jochen Schulz

Georg-August Universität Göttingen

Aufbau

1. **Vektoren**
   - Matrizen
   - Vektorräume
2. **Funktionen**

Aufbau

Aufbau

1. **Vektoren**
   - Matrizen
   - Vektorräume

2. **Funktionen**

Matrizen

$m \times n$ Matrix $A = (a_{ij}) \in K^{m \times n}$ über einen Körper K

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

$a_{ij} \in K$, Zeilenindex $i \in [1, m]$, Spaltenindex $j \in [1, n]$

Definitionen

- Transponiert von $A = (a_{ij})$: $A^T := (a_{ji})$.
- Symmetrisch: wenn $A = A^T$ gilt.
- Adjungiert von $A = (a_{ij}) \in \mathbb{C}^{m \times n}$: $A^* := (\overline{a_{ji}}) \in \mathbb{C}^{n \times m}$.
- Einheitsmatrix: $I := I_n := (\delta_{ij}) \in K^{n \times n}$
- Addition: Seien $A = (a_{ij}), B = (b_{ij}) \in K^{n \times m}$, dann

$$C = (c_{ij}) := A + B \in K^{n \times m}$$

  mit $c_{ij} = a_{ij} + b_{ij}$.

Definitionen

- Multiplikation: Seien $A = (a_{ij}) \in K^{m \times n}$ und $B = (b_{ij}) \in K^{n \times p}$, dann

$$C = (c_{ij}) := A \cdot B \in K^{m \times p}$$

  mit $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$.
- orthogonal: $A \cdot A^T = A^T \cdot A = I_n$ für $A \in K^{n \times n}$
- unitär: $A \cdot A^* = A^* \cdot A = I_n$ für $A \in \mathbb{C}^{n \times n}$.
- invertierbar: $A \in K^{n \times n}$ heißt , wenn eine Matrix $A^{-1} \in K^{n \times n}$ existiert mit $A \cdot A^{-1} = A^{-1} \cdot A = I_n$.

Definitionen und Bemerkungen

- Die Multiplikation ist assoziativ aber in der Regel nicht kommutativ.
- Die Matrizen aus $K^{m \times n}$ bilden einen Vektorraum über K (mit komponentenweiser Skalarmultiplikation).
- allgemeine lineare Gruppe $\mathrm{GL}(K, n) = \mathrm{GL}_n(K) = \mathrm{GL}(n, K)$: Die Menge der invertierbaren Matrizen aus $K^{n \times n}$ bilden bezüglich der Multiplikation eine Gruppe.
- orthogonale Gruppe: $O(n)$: Die Menge der orthogonalen Matrizen in $GL(\mathbb{R}, n)$ bilden eine Untergruppe von $GL(\mathbb{R}, n)$.
- unitäre Gruppe $U(n)$: Die entsprechende Untergruppe der unitären Matrizen in $GL(\mathbb{C}, n)$.

Matrizen in Sage

```
matrix([[a11,a12,...],[a21,a22,..],..])
```

- Der Rückgabewert ist vom Typ `matrix`.
- Die Einträge der Matrix können beliebige Ausdrücke sein.
- Es ist möglich Matrizen über bestimmten Bereichen (z.B. $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}$) zu konstruieren.
- Es gibt spezielle Datenstrukturen für quadratische Matrizen und für dünnbesetzte Matrizen.

Konstruktion von Matrizen I

Es gibt in Sage verschiedene Möglichkeiten eine Matrix zu konstruieren.
Beispiel:

$$A := \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & 0 & 1 & b \end{pmatrix}, \quad a, b \in \mathbb{R}$$

- Einträge pro Zeile in eckigen Klammern [..]. Alle Spalten dann wieder in eckigen Klammern [..] (Standard)

```
A = matrix([[1, 2, 3, 4],[a, 0, 1, b]])
```

- Explizite Größenangabe

```
A = matrix(2,4,[[1,2,3,4],[a,0,1,b]])
```

Konstruktion von Matrizen II

- Nullmatrix der Größe $n \times m$:

```
n = 3; m = 4; B = matrix(n,m)
```

- Erzeugung mit Hilfe einer Funktion $f(i,j)$ mit Einträgen $a_{ij} = f(i,j)$

```
f(i,j) = i*j
C = matrix([[f(i,j) for i in range(1,6)] for j in
    range(1,4)]); C
```

- Eingabe von Zeilen- und Spaltenvektoren (falls explizit nötig)

```
matrix(3,1,[1,2,3])
matrix(1,3,[4,5,6])
```

Zeilen- und Spaltenzahl

- Spaltenanzahl: `<matrix>.ncols()`

```
C.ncols()
```

```
5
```

- Zeilenanzahl: `<matrix>.nrows()`

```
C.nrows()
```

```
3
```

- Informationen über die Matrix: `<matrix>.parent()`

```
C.parent()
```

```
Full MatrixSpace of 3 by 5 dense matrices over
    Symbolic Ring
```

Zugriff auf die Einträge I

- Zugriff auf Einträge in Zeile i und Spalte j:

```
i=1; j=2; C[i,j]
```

6

- Ändern eines Eintrags in Zeile i und Spalte j:

```
i=1; j=2; C[i,j]=22
```

- Extrahieren von Zeilen/Spalten

```
zeile = C.row(0)
spalte = C.column(4)
```

Zugriff auf die Einträge II

- Teilmatrizen

```
C[1:3,1:3]
```

- Diagonalmatrizen

```
x = [1,2,3,4,5]
Diag = diagonal_matrix(x); Diag
```

Rechnen mit Matrizen

- Addieren, Multiplizieren

```
var('a,b,c,d,g,h,f')
A = matrix([[a, b], [c,d]])
B = matrix([[e, f], [g,h]])
A+B; A*B
```

- Inverse und der Transponierte

```
A^(-1); A.transpose()
```

Rang von Matrizen

Sei $A \in K^{m\times n}$.

- Spaltenrang: Die Dimension der linearen Hülle der Spaltenvektoren. Er ist höchstens gleich n.
- Zeilenrang: Die Dimension der linearen Hülle der Zeilenvektoren. Er ist höchstens gleich m.
- Rang: Abkürzung, da Zeilenrang und Spaltenrang von Matrizen gleich sind.

Matrizen

- Bestimmen des Ranges einer Matrix

```
S = matrix([[1,0,0],[0,1,1],[1,1,1]])
S.rank()
```

```
2
```

- ist S symmetrisch ? ist S invertierbar ?

```
S.is_symmetric()
S.is_invertible()
```

```
False
False
```

- Determinante

```
S.det()
```

```
27
```

Maps auf Matrizen

Anwendung der Funktion `<function>` auf `<matrix>` (siehe auch `map()`)

```
map_threaded(<function>,<matrix>)
```

Beispiel:

```
map_threaded(sqrt,matrix(RR,[[1,2],[3,4]]))
```

```
[1.00000000000000 1.41421356237310]
[1.73205080756888 2.00000000000000]
```

Aufbau

1. **Vektoren**
   - Matrizen
   - Vektorräume
2. **Funktionen**

Vektorraum

Ein Tripel $(V, +, \cdot)$, bestehend aus einer nichtleeren Menge V und Verknüpfungen

$$+ : V \times V \ \rightarrow \ V, \qquad \cdot : K \times V \ \rightarrow \ V$$

heißt Vektorraum über einem Körper K, wenn gilt:

1. $(V, +)$ ist eine abelsche Gruppe.
2. Für alle $v, w \in V$ und alle $\lambda, \mu \in K$ gilt:
   1. $(\lambda + \mu) \cdot v = (\lambda \cdot v) + (\mu \cdot v)$.
   2. $\lambda \cdot (v + w) = (\lambda \cdot v) + (\lambda \cdot w)$.
   3. $(\lambda\mu) \cdot v = \lambda \cdot (\mu \cdot v)$.
   4. $1 \cdot v = v$.

Begriffe

- Vektoren: Die Elemente eines Vektorraums.
- Skalarmultiplikation: Die Abbildung $\cdot : K \times V \rightarrow V$. Die Elemente des Körpers K nennt man Skalare.
- Untervektorraum oder Unterraum von V: Ist $U \subset V$ eine Teilmenge des Vektorraums V und es gelten alle Vektorraumaxiome.
- Vorsicht! man muß zwischen der 0 des Körpers und der 0 des Vektorraums (Nullvektor) unterscheiden.
  Es gilt $0 \cdot v = 0$ für alle $v \in V$.

Beispiele für Vektorräume

- $K^n := \{(x_1, \dots, x_n) \mid x_1, \dots, x_n \in K\}$, $n \in \mathbb{N}$
- Sei M eine beliebige Menge. Die Menge der Abbildungen von M in K, $\mathrm{Abb}(M, K)$, mit den punktweise definiertenVerknüpfungen

$$\begin{aligned} (f+g)(x) &:= f(x) + g(x), \forall\, x \in M \\ (\alpha \cdot f)(x) &:= \alpha \cdot f(x), \forall\, x \in M \end{aligned}$$

  für $\alpha \in K$, $f, g : M \mapsto K$.
- Die Menge der Polynome bis zum Grad n.
- Die Menge aller Polynome.
- $\mathbb{R}$ als $\mathbb{Q}$-Vektorraum.
- $\mathbb{C}$ als $\mathbb{R}$-Vektorraum.

Vektoren in Sage

Konstruktion von Vektoren

```
vector([v1,v2,..])
```

Beispiele:

- Konstruktion

```
a = vector([1,2,3,4]); b = vector([5,6,7]); a,b
```

```
((1, 2, 3, 4), (5, 6, 7))
```

- Datentyp: `vector_integer_dense` (für dichte Vektoren)

```
type(a)
```

```
<type 'sage.modules.vector_integer_dense.
    Vector_integer_dense'>
```

Lineare Abhängigkeit

Sei V ein K-Vektorraum und $(v_1, \ldots, v_r)$ eine Familie von Elementen aus V.

- Linearkombination $v \in V$ von $(v_1, \ldots, v_r)$: falls $\exists \lambda_1, \ldots, \lambda_r \in K$ mit $v = \lambda_1 v_1 + \cdots + \lambda_r v_r$.
- Lineare Hülle $span\{v_1, \ldots, v_n\}$: Die Menge aller Linearkombinationen. Die Lineare Hülle ist ein Unterraum von V.
- linear unabhängig: Sind $\lambda_1, \ldots, \lambda_r \in K$ und ist $\lambda_1 v_1 + \cdots + \lambda_r v_r = 0$ so folgt $\lambda_1 = \cdots = \lambda_r = 0$. Andernfalls linear abhängig.
    - Ist $M \subseteq V$ eine unendliche Menge, dann ist M linear unabhängig falls alle endlichen Teilmengen von M linear unabhängig sind.

Weitere Notationen und Bemerkungen

Sei V ein K-Vektorraum und $(v_1, \ldots, v_r)$ eine Familie von Elementen aus V

- $(v_1, \ldots, v_r)$ sind genau dann linear unabhängig, wenn sich jeder Vektor $v \in span\{v_1, \ldots, v_r\}$ eindeutig linear kombinieren läßt.
- Vektoren sind linear unabhängig wenn die Determinante der korrelierenden Matrix ungleich 0 ist.
- Gilt $V = span\{v_1, \ldots, v_r\}$, so ist $(v_1, \ldots, v_r)$ ein Erzeugendensystem. Sind $(v_1, \ldots, v_r)$ zusätzlich linear unabhängig, so ist $(v_1, \ldots, v_r)$ eine Basis.
- Aus jedem Erzeugendensystem kann man eine Basis auswählen.

Beispiele für Basen

- Seien $(e_i)_{i=1,\dots,n} \in \mathbb{R}^n$ die Einheitsvektoren. $(e_1, \dots, e_n)$ ist eine Basis des $\mathbb{R}^n$.
- Die Monombasis $(1, x, x^2, \dots, x^n)$ ist eine Basis des Vektorraums der Polynome n-ten Grades.
- $(1, i)$ ist eine Basis von $\mathbb{C}$ als $\mathbb{R}$-Vektorraum.
- $\mathbb{R}$ als $\mathbb{Q}$-Vektorraum hat keine endliche Basis.

Basis und Dimension

- Dimension des Vektorraums V: die Anzahl der Basiselemente einer Basis $(v_1, \ldots, v_n)$.
- Jeder Vektorraum besitzt eine Basis.
- Seien W, Z Unterräume von V. Dann ist $W + Z := span(W \cup Z)$ die Summe von W und Z. Es gilt:

$$\dim(W + Z) = \dim(W) + \dim(Z) - \dim(W \cap Z)$$

Sage

- Bestimmen einer Basis von $span(s1, s2, s3)$

```
s1 = vector([1,0,0])
s2 = vector([0,1,1])
s3 = vector([1,1,1])
span([s1,s2,s3],QQ)
```

```
Vector space of degree 3 and dimension 2 over
    Rational Field
Basis matrix:
[1 0 0]
[0 1 1]
```

Sage

- Bestimmen des Schnitts von *span*$(s1)$ und *span*$(s2, s3)$

```
span([s1],QQ).intersection(span([s2,s3],QQ))
```

```
Vector space of degree 3 and dimension 1 over
    Rational Field
Basis matrix:
[1 0 0]
```

- Testen der linearen Unabhängigkeit

```
m = matrix([s1,s2,s3])
m.rank() >= min(m.ncols(), m.nrows())
m.determinant() <> 0
```

```
False
```

Normen auf Vektorräumen

Sei V ein Vektorraum über $K = \mathbb{R}$ oder $K = \mathbb{C}$.
Eine Norm auf V ist eine Abbildung

$$\|\cdot\| : V \to \mathbb{R}, v \mapsto \|v\|,$$

so dass für alle $\alpha \in K$, $u, v \in V$ gilt

$$\begin{array}{rcl} \|v\| & \geq & 0 \\ \|v\| & = & 0 \text{ impliziert } v = 0 \\ \|\alpha v\| & = & |\alpha| \|v\| \\ \|u + v\| & \leq & \|u\| + \|v\| \text{ (Dreiecksungleichung).} \end{array}$$

$(V, \|\cdot\|)$ heißt normierter Raum.

Skalarprodukt

Eine skalarwertige binäre Abbildung

$$(\cdot,\cdot) : V \times V : \to \;\; K$$

auf einem Vektorraum V über $K = \mathbb{R}$ oder $K = \mathbb{C}$ heißt Skalarprodukt, wenn für alle $x, y, z \in V$, $\alpha, \beta \in K$ gilt

$$\begin{array}{rcl} (x,x) & \geq & 0 \\ (x,x) & = & 0 \text{ impliziert } x = 0. \\ (x,y) & = & \overline{(y,x)} \\ (\alpha x + \beta y, z) & = & \alpha(x,z) + \beta(y,z) \end{array}$$

Bemerkungen

- Ein VR V mit Skalarprodukt heißt Prä-Hilbert-Raum. Ist $K = \mathbb{R}$ so heißt der Raum auch euklidisch.
- Durch $\|v\| := \sqrt{(v, v)}$, $v \in V$ läßt sich eine Norm definieren. Es gilt die Cauchy-Schwarzsche Ungleichung

$$|(u, v)| \leq \|u\| \|v\|.$$

- Im euklidischen Raum ist der Winkel α zwischen zwei Vektoren $u, v \in V \setminus \{0\}$ definiert durch

$$\cos(\alpha) = \frac{(u, v)}{\|u\| \|v\|}.$$

Bemerkungen

- Orthogonal: wenn $(u, v) = 0$ gilt.
- Orthogonalbasis: Eine Basis aus paarweise orthogonalen Vektoren.
- Orthonormalbasis: Eine Orthogonalbasis, bei der alle Vektoren die Norm 1 haben.
- Jeder endlichdimensionale Prä-Hilbert-Raum hat eine Orthonormalbasis.
- Orthogonalraum:

  $$U^\perp := \{v \in V \mid (v, u) = 0 \text{ für alle } u \in U\}$$

  wenn U ein Unterraum von V ist.
- Es gilt: $\dim U + \dim U^\perp = \dim V$, insb. $U \cap U^\perp = 0$.

Sage

- p-Norm:

$$\|v\| := (\sum_{i=1}^{n} |v_i|^p)^{1/p}, p \in [1, \infty)$$

auf K^n mit $K = \mathbb{R}, \mathbb{C}$

```
x = vector([1,2,3,4,5]); p=2
x.norm(p)
```

- sind Vektoren orthogonal zueinander ?

```
a1 = vector([1,2,3])
a2 = vector([0,4,1])
a3 = vector([1,1,1])
X = matrix([a1,a2,a3])
X * X.transpose()
```

Sage

- Berechnen des Skalarprodukts

```
a2.dot_product(a3), a2*a3
```

```
(5, 5)
```

- Berechnen des Winkels zwischen zwei Vektoren

```
float(acos(a2*a3/(abs(a2)*abs(a3))))
```

```
0.79520271328967818
```

Aufbau

Aufbau von Funktionen

```
def <Name><(a,b,..)>:
    <Code-Block>
    return <ret>
```

- Argumente/Übergabeparameter `(a,b,..)`
- Jede Zeile, die einen folgenden Block einleitet, muss mit einem Doppelpunkt `:` abgeschlossen werden.
- Jede Zeile in diesem Block muss eine grössere Einrückung besitzen (typischerweise ein *Tab*)
- Zeilen gleicher Einrückung gehören zum gleichen Block.
- Rückgabewert `ret` mittels `return(ret)` zurückgegeben. Funktion ist dann beendet.
- Ohne return wird eine leere Variable des Typs `NoneType` zurückgegeben.

Erklärungen/Kommentare

- In `""" comment """` eingeschlossene Zeilen werden als Hilfetext abgespeichert und können durch `<funktionsname>?` abgefragt werden.
- Kommentarzeilen: Diese beginnen mit `#`. Sie werden vom System ignoriert.

Eine Beispiel-Funktion

```
def MyMax(a,b):
    """Maximum von a und b"""
    if a<b:
        return (b)
    else:
        return (a)
```

- Das Beispiel berechnet das Maximum zweier Zahlen a und b.
- Aufruf in Sage ist `MyMax(a,b)`.
- Die Funktion gibt dann entweder den Wert a oder den Wert b zurück.

Externes Editieren von Funktionen

Bei umfangreicheren Funktionen oder richtigen Programmen ist es evtl. sinnvoller dies mit einem Editor zu erstellen.

- Erstellen einer `<name>.sage`-Datei (Editor: z.B. `geany`)
- `attach <path>/<name>.sage`: hängt die Datei an das Notebook an und wird automatisch geladen und aktuell gehalten.
- `attached_files()`: listet alle angehängte Dateien.