Numerisk Lineær Algebra F2021
Notesæt 20

Andrew Swann

14. april 2021

Sidst ændret: 14. april 2021.
Versionskode: 83d371a.

Indhold

20 Lineære transformationer og koordinatskift

20.1 Generelle matrixrepræsentationer

Lad $L\colon V \to W$ være en lineær transformation. Antag at $E\colon v_0, v_1, \dots, v_{n-1}$ er en basis for V og at $F\colon w_0, w_1, \dots, w_{m-1}$ er en basis for W.

Definition 20.1. Vi definerer *matricen A af L mht. E og F* til at være matricen $A = \left[a_0 \,\middle|\, a_1 \,\middle|\, \dots \,\middle|\, a_{n-1} \right] \in \mathbb{R}^{m \times n}$ givet ved

$$a_j = [L(v_j)]_F. \tag{20.1}$$

Vi siger også at A er *matrixrepræsentationen* af L mht. E og F.

Denne matrixrepræsentation er nyttigt da det følgende resultat fortæller os at lineære transformationer er blot multiplikation med en matrix. Mere præcis kan vi erstatte L med matrixmultiplikation ved $A \in \mathbb{R}^{m \times n}$, hvis vi skriver vektorerne ud, som koordinatvektorer mht. de givne baser. Vi er glad for matrixmultiplikation, da det er en operation vi kan udføre i python.

Proposition 20.2. *Lad A være matricen af $L \colon V \to W$ mht. baserne E for V og F for W. Da gælder for alle $v \in V$ at*

$$[L(v)]_F = A[v]_E.$$

Bevis. Lad os skrive

$$[v]_E = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}.$$

Så har vi $v = x_0 v_0 + x_1 v_1 + \dots + x_{n-1} v_{n-1}$. Vi kan nu regne

$$\begin{aligned} L(v) &= L(x_0 v_0 + x_1 v_1 + \dots + x_{n-1} v_{n-1}) \\ &= x_0 L(v_0) + x_1 L(v_1) + \dots + x_{n-1} L(v_{n-1}). \end{aligned}$$

Skrives $L(v)$ ud i koordinater, får vi så

$$\begin{aligned} [L(v)]_F &= x_0 [L(v_0)]_F + x_1 [L(v_1)]_F + \dots + x_{n-1} [L(v_{n-1})]_F \\ &= x_0 a_0 + x_1 a_1 + \dots + x_{n-1} a_{n-1} \\ &= Ax. \end{aligned}$$

Det sidste udtryk er lige med $A[v]_E$, som ønsket. □

Eksempel 20.3. Lad V være vektorrummet af polynomier af grad højst 2 og lad W være vektorrummet af polynomier af grad højst 1. Funktionen $L \colon V \to W$ givet ved $L(p) = p'$, den afledede til p, er en lineær afbildning.

Lad os vælge baserne $E\colon x^2, x, 1$ for V, og $F\colon x, 1$ for W. Vi finder søjlerne a_0, a_1, a_2 af matricen $A \in \mathbb{R}^{2\times 3}$ af L mht. E og F ved

$$\begin{aligned}
a_0 &= [L(x^2)]_F = [2x]_F = [2\times x + 0\times 1]_F = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \\
a_1 &= [L(x)]_F = [1]_F = [0\times x + 1\times 1]_F = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
a_2 &= [L(1)]_F = [0]_F = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\end{aligned}$$

Dette giver

$$A = \begin{bmatrix} a_0 \mid a_1 \mid a_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Proposition 20.2 siger nu at L svarer til multiplikation med matricen A.

I dette tilfælde har vi for $p(x) = ax^2 + bx + c$ at $L(p(x)) = 2ax + b$ kan udføres, som matrixmultiplikation

$$[L(p(x))]_F = A[p(x)]_E = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2a \\ b \end{bmatrix},$$

og dermed implementeres f.eks. i python. △

Som konsekvens kan vi nu vise den følgende sætning. Formlen der fremkommer kaldes ofte *rang-nullitetsformlen*, da $\dim \operatorname{im} L$ er *rangen* af L, og $\dim \ker L$ en dens *nullitet*.

Sætning 20.4. *Lad $L\colon V \to W$ være lineær, lad E være en basis for V, lad F være en basis for W og lad A være matricen af L mht. E og F. Da gælder*

(a) *$v \in \ker L$ hvis og kun hvis $[v]_E \in N(A)$,*
(b) *$w \in \operatorname{im} L$ hvis og kun hvis $[w]_F \in S(A)$.*

Det følger at forskellige baser for V har det samme antal elementer og at

$$\dim \operatorname{im} L + \dim \ker L = \dim V.$$

Bevis. For del (a), har vi $v \in \ker L$ betyder at $L(v) = 0$. For sådan et $v \in \ker L$ har vi nu $A[v]_E = [L(v)]_F = [0]_F = 0$, så $[v]_E \in N(A)$. Omvendt $[v]_E \in N(A)$, giver $[L(v)]_F = A[v]_E = 0$, så $L(v) = 0$ og v ligger i $\ker L$.

Del (b) er tilsvarende: $w \in \operatorname{im} L$ betyder at $w = L(v)$ for et $v \in V$. Så for $w = L(v)$ har vi $[w]_F = [L(v)]_F = A[v]_E$, som siger at $[w]_F$ er en lineær kombination af søjle af A, dvs. $[w]_F \in S(A)$. Omvendt $[w]_F \in S(A)$ betyder $[w]_F = Ax$ for et $x \in \mathbb{R}^n$. Hvis basen E består af $v_0, \dots, v_{n-1}$, sætter vi $v = x_0 v_0 + \cdots + v_{n-1} x_{n-1}$ og får at $[v]_E = x$. Vi har nu $[L(v)]_F = Ax = [w]_F$, så $L(v) = w$ og $w \in \operatorname{im} L$.

Bemærk nu at

(a) $u_0, \dots, u_{k-1}$ er en basis for $\ker L$ hvis og kun hvis $[u_0]_E, \dots, [u_{k-1}]_E$ er en basis for $N(A)$, og

(b) $f_0, \dots, f_{r-1}$ er en basis for $\operatorname{im} L$ hvis og kun hvis $[f_0]_F, \dots, [f_{r-1}]_F$ er en basis for $S(A)$.

Ved at bruge proposition 19.11, har vi

$$\dim \operatorname{im} L + \dim \ker L = r + k = \dim S(A) + \dim N(A) = n = \dim V,$$

som er den ønskede relation mellem dimensioner.

For at se at forskellige baser for V har det samme antal elementer, antag nu at E og F er begge baser for V, hvor E har n elementer, og F har m elementer. Lad $L = \mathrm{Id}\colon V \to V$, $\mathrm{Id}(v) = v$ være identitetsafbildningen. Denne afbildning er lineær, så vi kan betragte matricen $A \in \mathbb{R}^{m\times n}$ af $L = \mathrm{Id}$ mht. E og F. Vi har $\ker \mathrm{Id} = \{0\}$ og $\operatorname{im} \mathrm{Id} = V$, så $N(A) = \{0\}$, $S(A) = \mathbb{R}^m$. Vi får nu at

$$n = \dim S(A) + \dim N(A) = m + 0 = m,$$

så både E og F har n elementer. □

Eksempel 20.5. I eksempel 20.3 kan vi godt beregne $\ker L$. Et polynomium $p(x) = ax^2+bx+c$ ligger i $\ker L$ hvis $L(p(x)) = p'(x) = 2ax+b$ er 0-polynomiet. Dette er det samme som $a = 0 = b$, så $\ker L$ består af alle konstante polynomier $p(x) = c$, med c en vilkårlig skalar. Bemærk at disse polynomier har $[p(x)]_E = [c]_E = (0, 0, c)$.

Matricen

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

af L mht. E og F har $v = (x_0, x_1, x_2) \in N(A)$ kun hvis $Av = 0$. Men

$$Av = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_0 \\ x_1 \end{bmatrix},$$

så $Av = 0$ kun for $v = (0, 0, x_2)$, med x_2 vilkårlig. Vektorer af denne form er netop koordinatvektorer for elementer af $\ker L$.

Bemærk at vi har $\dim \ker L = 1 = \dim N(A)$, da der er kun én fri parameter.

Billedmængden $\operatorname{im} L$ består af alle polynomier af formen $2ax + b$, så af alle polynomier af grad højst 1. Vi har dermed at $\dim \operatorname{im} L = 2$, og kan bekræfte at

$$\dim \operatorname{im} L + \dim \ker L = 2 + 1 = 3 = \dim V$$

i overenstemmelse med sætning 20.4. △

20.2 Basisskift

Vi fortsætter med at betragte en lineær afbildning $L\colon V \to W$. Hvis E og E' er to baser for V, har vi en invertibel koordinatskiftsmatrix S fra E til E', som opfylder

$$[v]_{E'} = S[v]_E$$

Tilsvarende hvis F og F' er baser for W, har vi en koordinatskiftsmatrix T fra F til F' med

$$[w]_{F'} = T[w]_F.$$

Vi kan nu beregne matricen B af L mht. E' og F', givet matricen A af L mht. E og F. Vi har

$$B[v]_{E'} = [L(v)]_{F'} \quad \text{og} \quad A[v]_E = [L(v)]_F,$$

så

$$BS[v]_E = B[v]_{E'} = [L(v)]_{F'} = T[L(v)]_F = TA[v]_E$$

for alle v. Det følger at

$$BS = TA,$$

som er det samme som

$$B = TAS^{-1}. \tag{20.2}$$

Omvendt, hver gang vi har en matrixformel af formen (20.2) med $A, B \in \mathbb{R}^{m\times n}$, og invertibel $S \in \mathbb{R}^{n\times n}$, $T \in \mathbb{R}^{m\times m}$, kan vi betragte dette som sigende at A og B er matrixrepræsentationer for den samme lineær transformation, blot i forhold til forskellige baser.

Vi kender allerede en formel af formen (20.2), nemlig den fulde SVD: $A = U\Sigma V^T$, som er $A = U\Sigma V^{-1}$, da V er ortogonal. Dette siger at A mht. standard baser har samme effekt som Σ mht. baser fra søjlerne af V og U: $Av_i = \sigma_i u_i$. Dette er et godt eksempel på at et nyt valg af baser kan gøre effekten af A mere tydeligt.

20.3 Afbildning på ét vektorrum

Betragt nu en lineær afbildning $L\colon V \to V$, som afbilder V ind i sig selv. Vælger vi en basis $E : v_0, \dots, v_{n-1}$ for V, har vi en *matrix A af L mht. kun E* givet ved (20.1) med $F = E$, dvs. $A = \begin{bmatrix} a_0 \mid a_1 \mid \dots \mid a_{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}$ med

$$a_j = [L(v_j)]_E.$$

Skifter vi til en anden basis $F\colon w_0, w_1, \dots, w_{n-1}$ af V, har vi en koordinatskiftsmatrix S fra E til F. Ved brug af (20.2), ser vi nu at matricen B af L mht. kun F er relateret til A via

$$B = SAS^{-1}.$$

Eksempel 20.6. I et lille land bor 500 000 mennesker i storbyen og 700 000 mennesker i landområder. Der observeres at hvert år flyttes 3 % af landbefolkningen til byen, og 2 % af bybefolkningen på landet. Lad os undersøge hvordan befolkningsgrupperne udvikler sig med tid.

Vi starter med befolkningsfordelingen

$$b_0 = \begin{bmatrix} 500\,000 \\ 700\,000 \end{bmatrix},$$

hvor vi skrive antallet i byen øverst, og antallet på landet nederst. Ændringen fra år til år gives ved matricen

$$A = \begin{bmatrix} 0{,}98 & 0{,}03 \\ 0{,}02 & 0{,}97 \end{bmatrix}.$$

Vi kan så beregne befolkningsfordeling over de kommende år, som

$$b_1 = Ab_0, \quad b_2 = Ab_1, \quad b_3 = Ab_2, \dots$$

I python

```python
import numpy as np
b0 = np.array([500000., 700000.])[:, np.newaxis]
a = np.array([[0.98, 0.03],
              [0.02, 0.97]])
b = b0
for i in range(5):
    print(f'b{i} = ', b)
    b = a @ b
```

```
b0 =  [[500000.]
 [700000.]]
b1 =  [[511000.]
 [689000.]]
b2 =  [[521450.]
 [678550.]]
b3 =  [[531377.5]
 [668622.5]]
b4 =  [[540808.625]
 [659191.375]]
```

Det ses at bybefolkningen vokser, men det er ikke klart hvad der sker over længere tid.

Lad os prøve at skifte fra standard basen, til basen F: $w_0 = (3,2), w_1 = (1,-1)$. Koordinatskiftsmatricen fra F til standardbasen er

$$T = \begin{bmatrix} w_0 \mid w_1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 2 & -1 \end{bmatrix}.$$

Effekten af A mht. F er givet ved matricen $C = T^{-1}AT$. Vi har at

$$\begin{aligned} C = T^{-1}AT &= \frac{1}{3\times(-1) - 1\times 2}\begin{bmatrix} -1 & -1 \\ -2 & 3 \end{bmatrix}\begin{bmatrix} 0{,}98 & 0{,}03 \\ 0{,}02 & 0{,}97 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 2 & -1 \end{bmatrix} \\ &= -\frac{1}{5}\begin{bmatrix} -1{,}0 & -1{,}0 \\ -1{,}9 & 2{,}85 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 2 & -1 \end{bmatrix} = -\frac{1}{5}\begin{bmatrix} -5 & 0 \\ 0 & -4{,}75 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0{,}95 \end{bmatrix} \end{aligned}$$

Bemærk at

$$A = TCT^{-1}$$

og at

$$\begin{aligned} A^2 &= (TCT^{-1})(TCT^{-1}) = TC(T^{-1}T)CT^{-1} = TC^2T^{-1} \\ A^3 &= A^2A = TC^2T^{-1}TCT^{-1} = TC^3T^{-1}. \end{aligned}$$

Så generelt har vi

$$A^k = TC^kT^{-1}.$$

Men matricen C^k er nem at udregne: det er en diagonalmatrix med indgange de k'te potenser af de tilsvarende indgange i C. Dette giver

$$\begin{aligned} b_k &= A^k b_0 = TC^kT^{-1}b_0 \\ &= T\begin{bmatrix}1 & 0\\ 0 & 0{,}95^k\end{bmatrix}T^{-1}b_0 \\ &= \begin{bmatrix}3 & 0{,}95^k\\ 2 & -0{,}95^k\end{bmatrix}\begin{bmatrix}240\,000\\ 220\,000\end{bmatrix} = \begin{bmatrix}720\,000\\ 480\,000\end{bmatrix} + 0{,}95^k\begin{bmatrix}240\,000\\ 220\,000\end{bmatrix}. \end{aligned} \tag{20.3}$$

Vi kan nu se hvad sker nå k bliver stort. For $k \to \infty$, har vi $(0{,}95)^k \to 0$, da $|0{,}95| < 1$. Sml.

```
for k in range(0, 1000, 57):
    print(0.95**k)
```

```
1.0
0.053733545982740286
0.0028872939638792646
0.00015514454297379496
8.336466433853638e-06
4.479479024570453e-07
2.4069829214547706e-08
1.2933572748966046e-09
6.949667260276838e-11
3.734302652948301e-12
2.006573233156666e-13
1.0782029509155957e-14
5.793566784174943e-16
3.113088872015411e-17
1.6727730405279714e-18
8.988402709189803e-20
4.8297875028563765e-21
2.595216088715975e-22
```

Så fra (20.3) ser vi at

$$b_k \to \begin{bmatrix}720\,000\\ 480\,000\end{bmatrix} \qquad \text{for } k \to \infty.$$

Dvs. efter mange år vil befolkningen blive fordelt med tæt på 720 000 i byen, og 480 000 på landet.

Alternativt kan vi se at

$$A^k = TC^kT^{-1} = T \begin{bmatrix} 1 & 0 \\ 0 & (0{,}95)^k \end{bmatrix} T^{-1} \to T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} T^{-1}.$$

Så har vi $b_k = A^k b_0 \to T \operatorname{diag}(1,0) T^{-1} b_0$ for $k \to \infty$. I python har vi

```
t = np.array([[3.0, 1.0],
              [2.0, -1.0]])
print(t)
```

```
[[ 3.  1.]
 [ 2. -1.]]
```

og matricen $C = TAT^{-1}$ er

```
c = np.linalg.inv(t) @ (a @ t)
print(c)
```

```
[[ 1.00000000e+00 -2.77555756e-17]
 [ 0.00000000e+00  9.50000000e-01]]
```

Grænsen for b_k når $k \to \infty$ er nu

```
print(t @ np.diag([1.0, 0.0]) @ np.linalg.inv(t) @ b0)
```

```
[[720000.]
 [480000.]]
```

som stemmer overens med grænsen fundet ovenfor. △

Python indeks

Indeks