

Numerieke Benadering

Pieter Vanderschueren

Academiejaar 2024-2025

Inhoudsopgave

Lineaire Benaderingsproblemen	2
1 Benadering van vectoren	3
1.1 QR-factorisatie	6
2 Benadering van functies	8
3 Benadering door middel van veeltermen	9
4 Discrete benadering op basis van meetdata	10
5 Regularisatietechnieken	11
Data, Grafen en Eigenwaarden	12
6 Grafen en eigenwaarden in datawetenschappen	13
7 Eigenwaardenalgoritmes	14
Niet-lineaire Benaderingsproblemen	15
8 Niet-lineaire benaderingsproblemen	15
9 Optimalisatie-algoritmes	16
10 Ijle representatie en benaderingen	17

Lineaire Benaderingsproblemen

1 Benadering van vectoren

Definitie 1.1: Orthogonale en orthonormale basissen

We spreken van een orthogonale, respectievelijk orthonormale basis als de basisvectoren $\{a_1, \dots, a_n\}$ orthogonaal, respectievelijk orthonormaal zijn. Als een basis niet orthogonaal is, spreken we van een scheve basis.

Definitie 1.2: Grammatrix

Voor een basis $\{a_1, \dots, a_n\}$ van deelruimte \mathcal{D} en twee vectoren $v, w \in \mathcal{D}$ ontbonden als

$$v = \sum_{i=1}^n \alpha_i a_i, \quad w = \sum_{i=1}^n \beta_i a_i$$

geldt dat het inwendig product ($\langle v, w \rangle = v^* w$) van v en w gelijk is aan

$$\langle v, w \rangle = \left(\sum_{i=1}^n \alpha_i a_i^* \right) \left(\sum_{i=1}^n \beta_i a_i \right) = [\alpha_1 \quad \dots \quad \alpha_n] \underbrace{\begin{bmatrix} \langle a_1, a_1 \rangle & \dots & \langle a_1, a_n \rangle \\ \vdots & \ddots & \vdots \\ \langle a_n, a_1 \rangle & \dots & \langle a_n, a_n \rangle \end{bmatrix}}_G \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$$

waarbij deze G de zogenaamde grammatrix is horende bij de basis $\{a_1, \dots, a_n\}$.

Eigenschap 1.1: Grammatrix

- Is de basis orthogonaal, dan is de grammatrix diagonaal.
- Is de basis orthonormaal, dan is de grammatrix de eenheidsmatrix.

Definitie 1.3: Projector

Een projector is een matrix $P \in \mathbb{C}^{m \times m}$ die idempotent is, dit is $P^2 = P$. De meetkundige betekenis is als volgt. Matrix P projecteert een vector op de ruimte $\mathcal{R}(P)$, waarbij de richting bepaald wordt door de nullspace $\mathcal{N}(P)$.

Toepassing 1.1: Projector

Stel $v \in \mathbb{C}^m$ een willekeurige vector en $P \in \mathbb{C}^{m \times m}$ een projector, dan is $Pv \in \mathcal{R}(P)$ volgens de definitie van het bereik, en is $(I - P)v \in \mathcal{N}(P)$, omdat

$$P(I - P)v = (P - P^2)v \stackrel{\text{idempotent}}{=} (P - P)v = 0$$

We kunnen dus v ontbinden in componenten volgens $\mathcal{R}(P)$ en $\mathcal{N}(P)$ als

$$v = Pv + (I - P)v$$

Deze ontbinding is uniek.

Eigenschap 1.2: Projector

- Als $v \in \mathcal{R}(P)$, dan is $Pv = v$.
- Er geldt dat $\mathcal{R}(P) \cap \mathcal{N}(P) = \{0\}$.
- Er geldt dat $\dim(\mathcal{R}(P)) + \dim(\mathcal{N}(P)) = m$.
- De ontbinding in componenten volgens $\mathcal{R}(P)$ en $\mathcal{N}(P)$ is uniek.

Bewijs 1.1: Projector

- Als $v \in \mathcal{R}(P)$, dan $\exists u : v = Pu$, en dus is $Pv = P^2u = Pu = v$.
- Stel $x \in \mathcal{R}(P)$ en $x \in \mathcal{N}(P)$. Er volgt dat $x = Px = 0$.
- Dit volgt uit de eerste dimensiestelling en vorige eigenschap.
- Stel $v = x_1 + y_1 = x_2 + y_2$, met $x_1, x_2 \in \mathcal{R}(P)$ en $y_1, y_2 \in \mathcal{N}(P)$. Er geldt voor $i \in 1, 2$ dat $Pv = Px_i + Py_i = x_i$. Hieruit volgt dat $x_1 = x_2$.

□

Definitie 1.4: Complementaire projector

Stel P een projector, dan is $\tilde{P} = I - P$ ook een projector. Hierbij geldt:

$$\mathcal{R}(P) = \mathcal{N}(I - P) = \mathcal{N}(\tilde{P}) \quad \text{en} \quad \mathcal{N}(P) = \mathcal{R}(I - P) = \mathcal{R}(\tilde{P}).$$

De ontbinding kan geschreven worden als

$$v = \underbrace{(I - \tilde{P})v}_{\in \mathcal{R}(P)} + \underbrace{\tilde{P}v}_{\in \mathcal{N}(P)}$$

Matrix \tilde{P} projecteert dus op $\mathcal{N}(P)$ waarbij de richting bepaald wordt door $\mathcal{R}(P)$. Dit is de **complementaire projector** van P .

Definitie 1.5: Orthogonale projector

Een projector P is orthogonaal indien $\mathcal{R}(P)$ en $\mathcal{N}(P)$ onderling orthogonale ruimte zijn. Een projector die niet orthogonaal is, noemen we een scheve projector.

Eigenschap 1.3: Orthogonale projector

Een projector P is orthogonaal als en alleen als $P = P^*$.

Bewijs 1.2: Orthogonale projector

“ \Rightarrow ”: Beschouw een orthonormale basis $\{q_1, \dots, q_n\}$ van $\mathcal{R}(P)$ en een orthonormale basis $\{q_{n+1}, \dots, q_m\}$ van $\mathcal{N}(P)$. Omdat volgens de definitie beide ruimten orthogonaal zijn, volgt dat

$$Q = [q_1 \quad \dots \quad q_n \quad q_{n+1} \quad \dots \quad q_m]$$

een unitaire matrix is. We verkrijgen:

$$PQ = [q_1 \quad \dots \quad q_n \quad 0 \quad \dots \quad 0] \Rightarrow Q^*PQ = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}.$$

Vermits Q^*PQ dus reëel is, geldt:

$$Q^*PQ = (Q^*PQ)^* = Q^*P^*Q$$

waaruit het gestelde volgt.

“ \Leftarrow ”: Neem willekeurige $x = Pu \in \mathcal{R}(P)$ en $y \in \mathcal{N}(P)$. Dan is:

$$\langle x, y \rangle = x^*y = (Pu)^*y = u^*P^*y = u^*Py = 0.$$

De ruimten $\mathcal{R}(P)$ en $\mathcal{N}(P)$ zijn dus orthogonaal. □

1.1 QR-factorisatie

Algoritme 1.1: Gram-Schmidt orthogonalisatie

```
1: for  $j = 1$  to  $n$  do
2:    $v_j = a_j$ 
3:   for  $i = 1$  to  $j - 1$  do
4:      $r_{ij} = q_i^* a_j$ 
5:      $v_j = v_j - r_{ij} q_i$  ( $= a_j - P_{\langle q_1, \dots, q_{j-1} \rangle} a_j$ )
6:   end for
7:    $r_{jj} = \|v_j\|_2$ 
8:    $q_j = v_j / r_{jj}$ 
9: end for
```

Algoritme 1.2: Gewijzigde Gram-Schmidt orthogonalisatie

```
1: for  $j = 1$  to  $n$  do
2:    $v_j = a_j$ 
3:   for  $i = 1$  to  $j - 1$  do
4:      $r_{ij} = q_i^* v_j$  ( $a_j \rightarrow v_j$ )
5:      $v_j = v_j - r_{ij} q_i$  ( $= (\mathbb{I} - P_{\langle q_{j-1} \rangle}) \dots (\mathbb{I} - P_{\langle q_2 \rangle}) (\mathbb{I} - P_{\langle q_1 \rangle}) a_j$ )
6:   end for
7:    $r_{jj} = \|v_j\|_2$ 
8:    $q_j = v_j / r_{jj}$ 
9: end for
```

Toepassing 1.2: Herorthogonalisatie van Gram-Schmidt

De Gram-Schmidt orthogonalisatie is numeriek instabiel. Dit kan verholpen worden door herorthogonalisatie, hieronder twee varianten waarvan de eerste de meest gebruikte is.

1. Stapsgewijze variant:

```
1:  $v_j = a_j$ 
2: for  $j = 1$  to  $j - 1$  do
3:    $r_{ij} = q_i^* v_j$ 
4:    $v_j = v_j - r_{ij} q_i$ 
5: end for
6:
7:  $w_j = v_j$ 
8: for  $i = 1$  to  $j - 1$  do
9:    $s_{ij} = q_i^* w_j$ 
10:   $v_j = v_j - s_{ij} q_i$ 
11:   $r_{ij} = r_{ij} + s_{ij}$ 
12: end for
```

2. **Simultane variant:** Na het berekenen van de onvolledige QR-factorisatie, die resulteert in factoren \hat{Q}_1 en \hat{R}_1 wordt het algoritme opnieuw toegepast met als input de eerste factor, wat resulteert in $\hat{Q}_1 \approx \hat{Q}_2 \hat{R}_2$. We bepalen dan $\hat{Q} = \hat{Q}_2$ en $\hat{R} = \hat{R}_2 \hat{R}_1$. Bij het gewijzigde algoritme van Gram-Schmidt (Algoritme 1.2) is dit meestal voldoende om orthogonaliteit van de kolommen van \hat{Q} te garanderen, bij het standaard algoritme (Algoritme 1.1) is soms meermaals herhalen van deze procedure noodzakelijk.

Algoritme 1.3: QR-facrotisatie met Givens-rotaties

```

1:  $Q = I, R = A$ 
2: for  $j = 1$  to  $n$  do
3:   for  $i = m$  downto  $j + 1$  do
4:      $c = \frac{r_{(i-1)j}}{\sqrt{r_{(i-1)j}^2 + r_{ij}^2}}, s = \frac{r_{ij}}{\sqrt{r_{(i-1)j}^2 + r_{ij}^2}}$ 
5:      $r_{ij} = 0, r_{(i-1)j} = \sqrt{r_{(i-1)j}^2 + r_{ij}^2}$ 
6:     for  $k = j + 1$  to  $n$  do
7:        $\begin{bmatrix} r_{(i-1)k} \\ r_{ik} \end{bmatrix} = \begin{bmatrix} \bar{c} & \bar{s} \\ -s & c \end{bmatrix} \begin{bmatrix} r_{(i-1)k} \\ r_{ik} \end{bmatrix}$ 
8:     end for
9:     for  $k = 1$  to  $m$  do
10:       $\begin{bmatrix} q_{k(i-1)} & q_{ki} \end{bmatrix} = \begin{bmatrix} q_{k(i-1)} & q_{ki} \end{bmatrix} \begin{bmatrix} \bar{c} & \bar{s} \\ -s & c \end{bmatrix}$ 
11:    end for
12:  end for
13: end for

```

Algoritme 1.4: QR-facrotisatie met Householder-rotaties

```

1:  $R = A$ 
2: for  $j = 1$  to  $n$  do
3:    $x = R(j : m, j)$ 
4:    $v_j = x + \text{sign}(x_1) \|x\|_2 e_1$ 
5:    $v_j = v_j / \|v_j\|_2$ 
6:    $R_{jj} = -\text{sign}(x_1) \|x\|_2, R(j + 1 : m, j) = 0$ 
7:   for  $k = j + 1$  to  $n$  do
8:      $R(j : m, k) = R(j : m, k) - 2(v_j^* R(j : m, k)) v_j$ 
9:   end for
10: end for
11: for  $j = 1$  to  $m$  do
12:    $w = e_i$ 
13:   for  $k = n$  downto 1 do
14:      $w_{k:m} = w_{k:m} - 2(v_k^* w_{k:m}) v_k$ 
15:   end for
16:    $Q(:, i) = w$ 
17: end for

```


2 Benadering van functies

3 Benadering door middel van veeltermen

4 Discrete benadering op basis van meetdata

5 Regularisatietechnieken

Data, Grafen en Eigenwaarden

6 Grafen en eigenwaarden in datawetenschappen

7 Eigenwaardenalgoritmes

Niet-lineaire Benaderingsproblemen

8 Niet-lineaire benaderingsproblemen

9 Optimalisatie-algoritmes

10 Ijle representatie en benaderingen