Samenvatting [G0Q57A] - Modellering en simulatie

Arne Van Den Kerchove

20 januari 2019

Inhoudsopgave

1 Modellen en simulaties

2 Numerieke lineaire algebra en toepassingen

2.1 QR-factorisatie

Definitie 1 *De volle QR-factorisatie van de matrix A wordt gegeven door*

$$A = QR$$

met q een $m \times m$ orthogonale matrix en R een $m \times n$ bovendriehoeksmatrix.

2.1.1 Gram-Schmidt orthogonalisatie

Algorithm 1 Gram-Schmidt-algoritme

procedure QRGramSchmidt
  for $j = 1$ to n **do**
    $v_j = a_j$
    for $i = 1$ to $j - 1$ **do**
      $r_{ij} = q_i^T a_j$
      $v_j = v_j - r_{ij} q_i$
    $r_{jj} = \|v_j\|_2$
    $q_j = v_j / r_{jj}$

Complexiteit: $\mathcal{O}(2mn^2)$
Stabiliteit: niet stabiel

2.1.2 QR-factorisatie met Givens-rotaties

Definitie 2 *Een Givens-rotatie is een $m \times m$ orthogonale matrix van de vorm*

$$G_{ij} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

met

$$c^2 + s^2 = 1$$

Om een Givens-rotatie op te stellen die plaats (j, k) 0 maakt in matrix A, kies dan een element in dezelfde kolom (bv. het element boven (j, k)) op oplaats (i, k) en maak G_{ij} met

$$c = \frac{a_{ik}}{\sqrt{a_{ik}^2 + a_{jk}^2}} \text{ en } s = \frac{a_{jk}}{\sqrt{a_{ik}^2 + a_{jk}^2}}$$

Algorithm 2 Givens-rotatie-algoritme

```
procedure QRGivens
    Q = 𝟙
    R = A
    for j = 1 to n do
        for i = m to j + 1 do
            c = r_{i-1,j} / sqrt(r_{i-1,j}^2 + r_{i,j}^2)
            s = r_{i,j} / sqrt(r_{i-1,j}^2 + r_{i,j}^2)
            r_{i,j} = 0
            r_{i-1,j} = sqrt(r_{i-1,j}^2 + r_{i,j}^2)
            for k = j + 1 to n do
```

$$\begin{bmatrix} r_{i-1,k} \\ r_{ik} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & s \end{bmatrix} \begin{bmatrix} r_{i-1,k} \\ r_{i,k} \end{bmatrix}$$

```
            for k = 1 to m do
```

$$\begin{bmatrix} q_{k,i-1} & q_{ki} \end{bmatrix} = \begin{bmatrix} q_{k,i-1} & q_{ki} \end{bmatrix} \begin{bmatrix} c & s \\ -s & s \end{bmatrix}$$

Complexiteit: $\mathcal{O}(3mn^2 - n^3)$
Stabiliteit: stabiel

2.1.3 QR-factorisatie met kolompivotering

Indien A niet van volle rang is, is het voor de stabiliteit beter om kolompivotering toe te passen. In stap j van het QR-algoritme met Givens-rotaties verwisselen we kolom j met de kolom p waarvan de 2-norm het grootst is.

Complexiteit: $\mathcal{O}(3mn^2 - n^3)$
Stabiliteit: stabieler voor rang-deficiënte matrices

2.2 Kleinste-Kwadratenbenadering

Om de coëfficiënten te bepalen wordt een Vandermondematrix A opgesteld. De te minimaliseren fout bij KK-benadering wordt gegeven door

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|_2 = \min_{x \in \mathbb{R}^n} \sqrt{\sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{i,j} x_j)^2}$$

met $r = b - Ax$ de *residuvector*.
Dit probleem kan opgelost worden door x te bepalen in

$$A^T A x = A^T b$$

De Vandermondematrix A is slecht geconditioneerd. We zoeken dus andere manieren om het KK-probleem op te lossen.

2.2.1 Oplossing met QR-ontbinding

Indien de QR-factorisatie van A bekend is, kan deze gebruikt worden om een oplossing voor het KK-probleem te vinden:

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2 = \min_{x\in\mathbb{R}^n} \|b - QRx\|_2 = \min_{x\in\mathbb{R}^n} \left\|Q^T b - RAx\right\|_2$$

Aangezien vermenigvuldiging vooraan met een orthogonale matrix de norm behoudt.
De vector $Q^T b = c$ kan opgesplitst worden in de volgende componenten: $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$
met $c_1 \in \mathbb{R}^n$ en $c_2 \in \mathbb{R}^{m-n}$
Hieruit volgt:

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2 = \min_{x\in\mathbb{R}^n} \left\| \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} - \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix} x \right\|_2$$

Volgens de stelling van Pythagoras geldt:

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2^2 = \min_{x\in\mathbb{R}^n} (\left\|c_1 - \hat{R}x\right\|_2^2 + \|c_2\|_2^2)$$

De vector x met coëfficiënten met minimale fout kan dus ook bekomen worden als x de oplossing van $\hat{R}x = c_1$

Complexiteit: $\mathcal{O}(mn) + \mathcal{O}n^2$ indien QR-factorisatie bekend
Stabiliteit: stabiel indien A van volle rang