

5.7. Ortogonaliseringsmetoder

Om man har problem med systemets kondition (vilket ofta är fallet), lönar det sig att undvika normalekvationerna vid lösning av minsta kvadratproblemet. En härtill lämplig metod är den så kallade **QR-metoden** eller ortogonaliseringsmetoden (jfr `qr`-kommandot i Matlab).

QR-teoremet: A må vara en given $m \times n$ matris med $m \geq n$ och linjärt oberoende kolonner. Då existerar det en entydig $m \times n$ matris Q , som har egenskapen

$$Q^T Q = D, \quad D = \text{diag}(d_1, \dots, d_n), \quad d_k > 0, \quad k = 1, 2, \dots, n,$$

och en entydig övre triangulär matris R , med diagonalelementen $r_{kk} = 1$, som har egenskapen

$$A = QR.$$

Låt oss se hur QR-metoden kan tillämpas på minsta kvadratmetoden: Av ekvationen $A^T(b - Ax) = 0$ följer att

$$R^T Q^T (b - Ax) = 0.$$

Eftersom R är icke-singulär och $Q^T A = Q^T Q R = D R$, så kan ovanstående ekvation också uttryckas $R^T(Q^T b - D R x) = 0$ eller

$$\begin{cases} R x = y \\ y = D^{-1} Q^T b. \end{cases}$$

Om man således känner Q och R , så kan man finna lösningen till minsta kvadratmetoden genom att lösa ett triangulärt ekvationssystem.

Den modifierade **Gram-Schmidt-metoden** för beräkning av Q , R och y : Vi beräknar en räkka matriser $A = A^{(1)}, A^{(2)}, \dots, A^{(n+1)} = Q$, där $A^{(k)}$ har formen

$$A^{(k)} = (q_1, q_2, \dots, q_{k-1}, a_k^{(k)}, \dots, a_n^{(k)}).$$

De $k - 1$ första kolonnerna i $A^{(k)}$ är lika med de $k - 1$ första kolonnerna i Q , och $a_k^{(k)}, \dots, a_n^{(k)}$ är vektorer, som har ortogonalerats mot q_1, \dots, q_{k-1} (dvs $a_i^{(k)T} q_j = 0$, $i = k, \dots, n$; $j = 1, \dots, k - 1$).

I det k :te steget ortogonaleras $a_j^{(k)}$, $j = k + 1, \dots, n$ mot q_k med följande procedur:

$$\begin{cases} q_k = a_k^{(k)}, & d_k = q_k^T q_k, & r_{kk} = 1 \\ a_j^{(k+1)} = a_j^{(k)} - r_{kj} q_k; & r_{kj} = q_k^T a_j^{(k)} / d_k, & j = k + 1, \dots, n. \end{cases}$$

Som vi ser, blir $q_k^T a_j^{(k+1)} = q_k^T a_j^{(k)} - r_{kj} q_k^T q_k = q_k^T a_j^{(k)} - \frac{q_k^T a_j^{(k)}}{q_k^T q_k} q_k^T q_k = 0$.

Vid varje steg kommer vi alltså att beräkna den k :te kolonnen av Q , samt den k :te raden av R ($r_{kj} = 0$, om $k > j$).

Vektorn b transformeras på analogt sätt: $b = b^{(1)}, b^{(2)}, \dots, b^{(n+1)}$, där

$$b^{(k+1)} = b^{(k)} - y_k q_k, \quad y_k = \frac{q_k^T b^{(k)}}{d_k}.$$

Som vi ser blir även $q_k^T b^{(k+1)} = q_k^T b^{(k)} - \frac{q_k^T b^{(k)}}{q_k^T q_k} q_k^T q_k = 0$. Här kommer $b^{(n+1)}$ att vara den del av b som är ortogonal mot $\mathcal{R}(A)$ (det underrum, som spännes av A :s kolonner), och den kommer därför att bli lika stor som restvektorn r .

Efter n steg ($k = 1, 2, \dots, n$) fås slutligen

$$Q = (q_1, q_2, \dots, q_n) \quad R = (r_{kj}), \quad y = (y_1, y_2, \dots, y_n)^T$$

och då blir $Q^T Q = \text{diag}(d_k)$, $A = QR$, $b = Qy + r$. Slutligen löses x ur ekvationen $Rx = y$.

Vid beräkningen av R och y krävs det approximativt

$$2m \sum_{k=1}^n (n - k + 1) = 2m \cdot \frac{n(n + 1)}{2} = mn(n + 1)$$

räkneoperationer, men för att lösa $Rx = y$ endast $n(n + 1)/2$ räkneoperationer. Gram–Schmidt–metoden kräver sålunda omkring dubbelt mera arbete än vad som behövs för att ställa upp normalekvationerna.

Tidigare har vi antagit, att A :s kolonner är linjärt oberoende. Av relationen $Q = AR^{-1}$, där $S = R^{-1}$ är en högertriangulär matris med diagonalelementen 1, framgår att q_k kan uttryckas som en linjär kombination av a_1, a_2, \dots, a_k . Antag nu, att a_1, a_2, \dots, a_{k-1} är linjärt oberoende, men att a_k beror linjärt av a_1, a_2, \dots, a_{k-1} och därför även av q_1, q_2, \dots, q_{k-1} . Vi finner då att $a_k^{(k)} = 0$, och ortogonaliseringsproceduren kan inte fortsättas.

Maximalantalet linjärt oberoende kolonner (eller rader) i en matris brukar kallas matrisens *rang*. Om rangen för $A > k - 1$, så måste det existera en vektor $a_j^{(k)} \neq 0$ om $k \leq j \leq n$. Då kan vi låta den k :te och j :te kolonnen byta plats, och fortsätta ortogonaliseringsprocessen ända tills alla de återstående kolonnerna beror linjärt av de beräknade q -vektorer.

Därav följer att vi kan förbättra Gram-Schmidt-metoden genom **kolonnpivotering**. I det k :te steget väljs s som det minsta heltal för vilket gäller $\|a_s^{(k)}\|_2 = \max_{k \leq j \leq n} \|a_j^{(k)}\|_2$, varpå kolonnerna k och s kastas om.

Ett annat sätt att utföra en ortogonalisering är att använda **Householder-transformationer** (denna metod utnyttjas i MATLAB-rutinen `qr`). De baserar sig på *elementära ortogonala matriser* som är n -dimensionella enhetsmatriser modifierade av enkla rotationsmatriser, såsom t.ex.

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix},$$

som representerar en rotation i planet omfattande vinkeln θ . Ett enkelt exempel på en dylik matris är

$$\begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & 1 & \dots & & 0 \\ \vdots & \vdots & \cos \theta & & \sin \theta \\ & & & 1 & \\ & & & & 1 \\ 0 & 0 & -\sin \theta & & \cos \theta \end{pmatrix},$$

som också representerar en plan rotation, men i ett n -dimensionellt rum.

Ett sätt att konstruera sådana matriser är att använda matriser av formen $P = I - 2uu^T/u^T u$, där $u \neq 0$ är en godtycklig vektor. På grund av sin konstruktion är Householders matris P symmetrisk ($P = P^T$) och ortogonal, emedan

$$\begin{aligned} P^T P &= P^2 = I - 4\frac{uu^T}{u^T u} + 4\frac{uu^T uu^T}{u^T u u^T u} \\ &= I - 4\frac{uu^T}{u^T u} + 4\frac{u(u^T u)u^T}{(u^T u)^2} = I - 4\frac{uu^T}{u^T u} + 4\frac{uu^T}{u^T u} = I, \end{aligned}$$

dvs $P^2 = I$ eller alltså $P^{-1} = P = P^T$.

Hur skall Householder-transformationen konstrueras, så att vi kan överföra matrisen i triangulär form? Låt oss beteckna den första kolonnen i matrisen A med a , och anta att P nollställer alla komponenter i a med undantag av den första:

$$Pa = \alpha(1, 0, \dots, 0)^T = \alpha e_1$$

Genom att använda den allmänna formen av P får vi då

$$Pa = \left(I - 2\frac{uu^T}{u^T u} \right) a = a - \left(2\frac{u^T a}{u^T u} \right) u = \alpha e_1.$$

Om denna ekvation skrivs i formen

$$\left(\begin{array}{c} u^T a \\ 2 \frac{u^T a}{u^T u} \end{array} \right) u = a - \alpha e_1,$$

så ser vi, att u är en multipel av $a - \alpha e_1$. Om u multipliceras med en godtycklig konstant som skiljer sig från noll, så förändras inte Householders matris, och vi kan därför välja $u = a - \alpha e_1$. Eftersom en ortogonal transformation skall bevara normen (eftersom det är fråga om en rotation), så är $\alpha = \pm \|a\|_2$, och vi får $u = a \mp \|a\|_2 e_1$.

Vi skall nu visa hur dylika matriser kan användas för att överföra en matris i triangulär form. Antag, att A är en 5×3 matris:

$$A = \begin{pmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{pmatrix}$$

Om vi betecknar den första kolonnen i A med a , så får vi

$$PA = \begin{pmatrix} \|a\| & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{pmatrix}$$

Observera, att P har förändrat elementen som betecknas med x .

I det följande steget väljer vi de fyra lägsta elementen i den andra kolonnen av PA och bildar en ny vektor a' med fyra element på vilken vi tillämpar en Householder-transformation P' . Vi får då

$$P_1PA = \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} \begin{pmatrix} \|a\| & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{pmatrix} = \begin{pmatrix} \|a\| & x & x \\ 0 & \|a'\| & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{pmatrix}.$$

Matrisen blir fullständigt reducerad till en triangulär matris, om vi bildar en ny vektor a'' av de tre nedersta elementen i matrisen P_1PA , och konstruerar en ny Householder-transformation P'' som nollställer denna vektor utom det översta elementet:

$$P_2P_1PA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & P'' \end{pmatrix} \begin{pmatrix} \|a\| & x & x \\ 0 & \|a'\| & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{pmatrix} = \begin{pmatrix} \|a\| & x & x \\ 0 & \|a'\| & x \\ 0 & 0 & \|a''\| \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Eftersom både P' och P'' är ortogonala matriser, så är också P_1 och P_2 ortogonala, liksom även P_2P_1P . Således har A blivit transformerad till en (övre) triangulär matris genom att den multiplicerats med en ortogonal matris. Vi får alltså $A = QR$, där $Q = PP_1P_2$ och $R = P_2P_1PA$. Uppenbarligen går metoden att generalisera till godtyckligt stora matriser.

Hur kan detta tillämpas på ett minsta kvadrat-problem $Ax = b$? Antag att A är en $m \times n$ matris som kan faktoriseras $A = QR = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, där R_1 är en $m \times m$ matris. Då får vi $Q^Tb = c \equiv \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$, där c_1 är en kolonnvektor med m element och c_2 en kolonnvektor med $n - m$ element.

Normen bevaras vid multiplikation med en ortogonal matris, varav följer

$$\|Ax - b\|_2^2 = \|Q^T(Ax - b)\|_2^2 = \|Rx - c\|_2^2 = \left\| \begin{pmatrix} R_1x - c_1 \\ -c_2 \end{pmatrix} \right\|_2^2 = \|R_1x - c_1\|_2^2 + \|c_2\|_2^2.$$

Vi finner således lösningen till minsta kvadratproblemet genom att lösa ekvationen $R_1x = c_1$, och det minsta värdet av kvadraternas summa blir $\|c_2\|_2^2$.

Det finns också fall då lösningen inte är unik (**degenererat minsta kvadrat-problem**). Detta innebär vanligen att modellfunktionerna inte är linjärt oberoende, vilket innebär att kolonnerna i matrisen A är linjärt beroende. Ett dylikt problem har många lösningar istället för en enda. I detta fall används ofta kolonnpivoting (se ovan) vid ortogonaliseringen, varvid A uttrycks som $A = QRP$, där P är en permutationsmatris, som håller reda på kolonnbyten som gjorts. Matrisen R har då formen

$$R = \begin{pmatrix} R_1 & R_2 \\ 0 & 0 \end{pmatrix},$$

där R_1 är en övre triangulär matris och $R_2 \neq 0$ är en rektangulär matris. Antalet rader olika noll i R är lika med matrisens rang (alltså antalet linjärt oberoende kolonner).

Faktoriseringen leder till ett reducerat minsta kvadrat-problem

$$\|Ax - b\|_2 = \|RPx - Q^T b\|_2.$$

Om vi nu inför beteckningarna $y = Px$ och $c = Q^T b$, som uppdelas på motsvarande sätt som R i två delar: $y = (y_1, y_2)^T$ samt $c = (c_1, c_2)^T$, så får vi

$$\begin{aligned}\|Ax - b\|_2^2 &= \left\| \begin{pmatrix} R_1 & R_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \right\|_2^2 \\ &= \|R_1 y_1 + R_2 y_2 - c_1\|_2^2 + \|-c_2\|_2^2.\end{aligned}$$

Den andra termen $-c_2$ påverkas inte av parametrarna y , medan den första termen kan minimeras på ett godtyckligt antal sätt. Komponenterna av y_2 kan väljas godtyckligt, sedan kan y_1 lösas ur ekvationssystemet

$$R_1 y_1 = c_1 - R_2 y_2.$$

Systemet är högertriangulärt och kan lösas genom bakåtsubstitution. Vanligen sätter man $y_2 = 0$.

5.8. Singulärvärdesuppdelningen

Antag, att A är en $m \times n$ -matris ($m \neq n$) med reella element. Det existerar då en $m \times m$ ortogonal matris U , en $n \times n$ ortogonal matris V , och en $m \times n$ diagonal matris D , vars diagonalelement $d_1 \geq d_2 \geq \dots d_s \geq 0$ ($s = \min(m, n)$), så att

$$A = UDV^T.$$

Matrisen U bildas av de ortonormerade egenvektorer, som motsvarar egenvärdena av matrisen AA^T , och matrisen V består av de ortonormerade egenvektorerna av $A^T A$. Diagonalelementen av matrisen D är de icke-negativa kvadratrötterna av egenvärdena av $A^T A$, som även brukar kallas **singulära värden**. Vi skall avstå från att bevisa riktigheten av detta.

Man kan också visa, att matrisens rang är lika med antalet singulära värden, som skiljer sig från noll. Om A är singulär så är åtminstone $d_n = 0$. Om matrisen är "nästan singulär", så betyder det, att några av de singulära värdena är mycket små. Förhållandet d_1/d_n kan uppfattas som ett mått på konditionen för matrisen A . Om man väljer $Q = U$ samt $R = DV^T$, så ser man att singulärvärdesuppdelningen leder till en ortogonal faktorisering.

Lösningen till ekvationssystemet $Ax = b$ kan då skrivas $x = VD^+U^Tb$, som kan beräknas i två steg:

$$\begin{cases} y = Q^Tb = U^Tb \\ x = A^+b = VD^+y, \end{cases}$$

där D^+ är en diagonal matris med diagonalelementen $1/d_k$ om $d_k > 0$, och 0 i annat fall.

Dessa ekvationer gäller oberoende av antalet singularvärden $d_k \neq 0$. För det fall att matrisens rang är okänd, kan denna metod vara nyttig att använda. För välartade matriser är den emellertid mer kostsam och något mindre noggrann.

Som vi lätt inser, gäller följande ekvationer:

$$A^T A = VD^T DV^T, \quad AA^T = UDD^T U^T.$$

Således är kvadraterna på de singulara värdena egenvärden för matriserna $A^T A$ och AA^T , och man skulle därför vänta sig att singularvärdesuppdelningen enkelt skulle kunna utföras genom diagonalisering av de symmetriska matriserna $A^T A$ och AA^T . Men detta leder emellertid inte till en stabil metod att utföra singularvärdesuppdelningen.

Singulärvärdesuppdelningen, som upptäcktes redan på 1870-talet av Beltrami och Jordan, infördes som en praktisk metod vid ekvationslösning av *Gene Golub*¹ på 1970-talet.

Singulärvärdesuppdelningen är mycket lämplig att använda, när det gäller att studera en matris med dålig kondition. Om matrisen A är kvadratisk, så är alla matriserna U , V och D kvadratiska matriser, och man finner, att inversen av matrisen A kan uttryckas

$$A^{-1} = V \text{diag}(1/d_j) U^T,$$

där $\text{diag}(1/d_j)$ betecknar en diagonalmatris, vars element består av de reciproka värdena av matrisen D :s element.

Detta uttryck är korrekt, såvida inte något av D :s element är mycket litet. Detta kan inträffa, om matrisen har dålig kondition. I detta fall blir konditionstalet d_1/d_n mycket stort. Om A är singulär, blir (åtminstone) ett av dess singulära värden noll. I detta fall kan man konstruera en "invers" matris genom att *ersätta* $1/d_j$ med noll då $d_j = 0$.

¹G.H. Golub och C. Reinsch: *Singular Value Decomposition and Least Squares Solutions*, Numer. Math. **14**, 403-420 (1970)

Detta är ett specialfall av en **pseudoinvers**. En dylik invers kan definieras för en godtycklig $m \times n$ -matris A som en matris X , som uppfyller *Penrose's villkor*:

$$\begin{aligned}AXA &= A, \\XAX &= X, \\(AX)^T &= AX, \\(XA)^T &= XA.\end{aligned}$$

Man kan lätt visa, att om A^+ är en pseudoinvers, och A har singularvärdessuppdelningen

$$A = UDV^T,$$

så kan dess pseudoinvers uttryckas som

$$A^+ = VD^+U^T,$$

där $D^+ = \text{diag}(d_i^+)$, och

$$d_i^+ = \begin{cases} 1/d_i & \text{om } d_i > 0, \\ 0 & \text{om } d_i = 0. \end{cases}$$

Det är lätt att visa, att de nämnda fyra villkoren är uppfyllda, och att A^+ inte heller beror av valet av U och V .

Singulärvärdesuppdelningen kan användas i minsta kvadratmetoden på följande sätt. För normen av rest-termen finner vi följande uttryck (observera, att U är ortogonal):

$$\begin{aligned}\|Ax - b\|_2 &= \|UDV^T x - b\|_2 \\ &= \|U^T(UDV^T x - b)\|_2 = \|DV^T x - U^T b\|_2.\end{aligned}$$

Genom att beteckna $U^T b$ med c och $V^T x$ med z , får man

$$\|Ax - b\|_2^2 = \|Dz - c\|_2^2 = (d_1 z_1 - c_1)^2 + \dots + (d_n z_n - c_n)^2 + c_{n+1}^2 + \dots + c_m^2.$$

Om inga singulärvärden är noll, så kan man alltid välja z_i (dvs x_i) så, att normen minimeras:

$$\|Ax - b\|_2^2 = c_{n+1}^2 + \dots + c_m^2.$$

I detta fall existerar det en entydig lösning.

Om $d_n = 0$, kan man välja vilket värde som helst av z_n , och man får alltid

$$\|Ax - b\|_2^2 = c_n^2 + c_{n+1}^2 + \dots + c_m^2,$$

som inte ger en entydig lösning. Om $d_i = 0$ för något i brukar man vanligen också välja $z_i = 0$. Små singulära värden visar att systemet har dålig kondition.

Om konditionen är dålig, så är det oförmånligt att lösa normalekvationerna, och singularvärdesuppdelningen är då att föredra. Om A är en $m \times n$ matris med rangen $r > 0$, och singularvärdena $d_1 \geq d_2 \geq \dots \geq d_r > 0$, så är konditionstalet för A : $\text{cond}(A) = \|A\|_2 \|A^+\|_2 = d_1/d_r$. Eftersom $A^T A = V D^T D V^T$, så är singularvärdena för $A^T A$ kvadraterna på singularvärdena för A , och $\text{cond}(A^T A) = (\text{cond}(A))^2$. Löser man minsta kvadratproblemet med singularvärdesuppdelning, visar sig felet bestämmas av $\text{cond}(A)$, istället för $\text{cond}(A^T A)$, som gäller för lösningen av ett system av normalekvationer.

I MATLAB kan singularvärdesuppdelningen utföras med den inbyggda funktionen `svd`, som alstrar de tre matriserna U , D och V . Nedan visas som exempel singularvärdesuppdelningen av en 2×3 matris. Som vi ser, är matrisens rang 2. Vi kan också beräkna matrisens pseudoinvers med funktionen `pinv`.

```

>> A = [1 1; 1 2 ; 1 3];
>> [U, S, V] = svd(A);
U =
    0.3231    -0.8538    0.4082
    0.5475    -0.1832   -0.8165
    0.7719     0.4873    0.4082
S =
    4.0791         0
         0    0.6005
         0         0
V =
    0.4027   -0.9153
    0.9153    0.4027

>> pseudo=pinv(A)
pseudo =
    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000

>> pseudo2 = V* pinv(S) * U'
pseudo2 =
    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000

```