(Trefethen-Bau, lekcia 27)

Rayleighov podiel, inverzné iterovanie

V tejto lekcii sa predstavia dva klasické algoritmy pre vlastné vektory / vl. hodnoty.
Samostatne sú oba užitočné pre niektoré úlohy, špeciálne inverzné iterovanie pre hľadanie vl. vektora, ak poznáme vl. hodnotu.
Spolu však budú základom pre QR algoritmus, ktorý popíšeme v nasledujúcich dvoch lekciách (častiach tejto prednášky)

Obmedzenie sa na reálne symetrické matice

~~Všeobecné~~ algoritmy numerickej lineárnej algebry fungujú pre všeobecné matice alebo pre hermitovské, kde si môžeme dovoliť vďaka ich štruktúre isté zjednodušenie.
V prípade hľadačov vl. hodnôt (QR algoritmus) sú však tieto rozdiely dosť veľké, preto pre zjednodušenie výkladu budeme uvažovať A reálnu & symetrickú, všetky normy budú $\|.\|_2$.
Takže v nasledujúcom budeme mať $A = A^T \in M_{m \times m}(\mathbb{R})$, $x \in \mathbb{R}^m$, $x^* = x^T$

$\|x\| = \sqrt{x^T x}$. Tiež to znamená, že A má reálne vl. hodnoty.
a je diagonalizovateľná ortogonálne.
(reálne) vlastné hodnoty označíme $\lambda_1, \lambda_2, \dots, \lambda_m$
ortonormálne vl. vektory $q_1, q_2, \dots, q_m$

Myšlienky z nasledujúcich lekcií sa týkajú fázy 2 pri hľadaní vl. hodnôt, teda matica A už prešla úpravami vo fáze 1 a je tridiagonálna.
Ako uvidíme, tridiagonálna forma má niektoré matematické výhody, ale hlavne algoritmické. Niektoré kroky budú totiž $O(m)$ namiesto $O(m^2)$

Rayleighov podiel

Rayleighov podiel pre vektor $x \in \mathbb{R}^n$ je podiel

$$r(x) = \frac{x^T A x}{x^T x}$$

Ak je x vl. vektor, potom $r(x) = \lambda$ je vl. hodnota.

Pozrime sa na motiváciu v pozadí:

Môžeme sa pýtať, "ktorý skalár α sa pre vektor x správa "najviac" ako vl. hodnota?"

t.j. kedy je $\|Ax - \alpha x\|_2$ minimálne? Toto je úloha najmenších $\underbrace{\text{štvorcov v}}_{\text{typu } m\times 1}$ tvare $x\alpha \approx Ax$ (x je matica, α neznámy vektor 1×1 a Ax pravá strana)

Riešením je práve $r(x)$. ~~[illegible]~~

$R(x)$ je teda prirodzený odhad pre veľkosť vl. hodnoty, ak x je blízko k vl. vektoru.

Môžeme to kvantifikovať ešte trochu lepšie:

$x \in \mathbb{R}^n$ - premenná

$r : \mathbb{R}^n \to \mathbb{R}$ funkcia.

Pozrime sa na parciálne derivácie podľa súradníc x_j:

$$\frac{\partial r(x)}{\partial x_j} = \frac{\frac{\partial}{\partial x_j}(x^T A x)}{x^T x} - \frac{(x^T A x)\frac{\partial}{\partial x_j}(x^T x)}{(x^T x)^2}$$

$$= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x)\, 2x_j}{(x^T x)^2} = \frac{2}{x^T x}\left(Ax - r(x)\,x\right)_j$$

Spojením parciálnych derivácií dostaneme gradient:

$$\nabla r(x) = \frac{2}{x^T x}\left(Ax - r(x)\,x\right)$$

Pre vl. vektor bude takýto gradient nulový ($r(x) = \lambda$)

Položme, ak $\nabla r(x) = \vec{0}$ (pre nenulové x), tak x je vl. vektor.

Geometrický význam:
$r(x)$ nezávisí od veľkosti x, iba od smeru (je to funkcia na projektívnej sfére)

takže sa stačí pozerať na jej hodnoty na 1-tkovej sfére.
V prípade jednoduchých vl. hodnôt (násobnosť = 1) budú vlastné hodnoty zodpovedať hodnotám r $\underbrace{}_{\text{izolovaných}}$ stacionárnych (kritických) bodoch $r(x)$.

Vieme povedať ešte viac: ($\nabla r(q_j) = \vec{0}$, z hladkosti $r(x)$)
ak q_j je vl. vektor, potom
(mimo $\vec{0}$) máme $r(x) - r(q_j) = O(\|x - q_j\|^2)$ pre $x \to q_j$

(Taylorov rozvoj)
Teda Rayleighov podiel je <u>kvadraticky presný</u> odhad vl. hodnoty.

Ešte inak: ak x je lin. kombinácia vl. vektorov $q_1, \dots, q_m$
tak $x = \sum_{j=1}^{m} a_j q_j$ a $r(x) = \sum_{j=1}^{m} a_j^2 \lambda_j \Big/ \sum_{j=1}^{m} a_j^2$.
Teda $r(x)$ je vážený priemer vl. hodnôt A vážený štvorcami súradníc vzhľadom na ~~[illegible]~~ vlastnú bázu.
vďaka to tiež $r(x) - r(q_J) = O(\varepsilon^2)$ pre $|a_j/a_J| < \varepsilon$ $j \neq J$.

<u>Mocninové iterovanie</u>

Zmeňme tému. Predpokladajme, že $v^{(0)}$ spĺňa $\|v^{(0)}\| = 1$.
Pomocou mocninového iterovania, ktoré sme už spomenuli ako nie veľmi dobrého kandidáta na hľadanie vl. vektorov (lekcia 25), dostaneme postupnosť $v^{(i)}$, ktorá bude konvergovať ku vl. vektoru pre najväčšiu vl. hodnotu matice A.

Algoritmus 27.1

$v^{(0)}$ = nejaký vektor s $\|v^{(0)}\| = 1$
pre $k = 1, 2, \ldots$
$w = A v^{(k-1)}$ — prenásobiť A
$v^{(k)} = w / \|w\|$ — normalizovať
$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — odhadnúť λ pomocou Rayleigh. podielu

– nebudeme špecifikovať, kedy treba takýto algoritmus ukončiť, preto len "pre $k=1,2,\ldots$". V praxi sú však ukončovacie podmienky veľmi dôležité – preto sú spravidla numerické knižnice výrazne lepšie ako ad-hoc naprogramované programy.

Mocninové iterovanie sa dá analyzovať pomerne ľahko.
$v^{(0)}$ napíšme ako kombináciu ortonormálnych vl. vektorov q_i:

$$v^{(0)} = a_1 q_1 + a_2 q_2 + \cdots + a_m q_m$$

$v^{(k)}$ je nejaký násobok $A^k v^{(0)}$, preto existujú konštanty c_k:

$$\begin{aligned} v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \cdots + a_m \lambda_m^k q_m) \\ &= c_k \lambda_1^k \left(a_1 q_1 + a_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k q_2 + \cdots + a_m \left(\frac{\lambda_m}{\lambda_1}\right)^k q_m \right). \end{aligned}$$

Z toho máme

<u>Tvrdenie</u> Nech $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_m| \geq 0$ a $q_1^T v^{(0)} \neq 0$.
Potom priebežné hodnoty Algoritmu 27.1 spĺňajú odhady:

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

~~Dôkaz~~ Pre $k \to \infty$. Znamienko $\pm$ znamená, že v každom kroku musíme vybrať jedno z nich, pre vhodné platí odhad.

<u>Dôkaz</u> Prvý vzťah plynie z vyjadrenia $v^{(k)}$ ako LK $q_1, \ldots, q_m$, keďže predpokladáme, že $a_1 = q_1^T v^{(0)} \neq 0$.
Druhý odhad vyplýva z kvadratickej presnosti Rayleighovho podielu.

Ak $\lambda_1 > 0$, potom sú všetky znamienka + alebo −, ak $\lambda_1 < 0$, tak alternujú.

Z výpočtového hľadiska ± znamienko nevyzerá veľmi vábne.
Mohli by sme hovoriť o "projektivizácii", t.j. konvergencii podpriestoru $\mathrm{span}\langle v^{(k)}\rangle$ a $\mathrm{span}\langle q_1\rangle$, ale to by si vyžadovalo definíciu konvergencie podpriestorov.

Mocninové iterovanie ~~má~~ samo o sebe limitovanú použiteľnosť
- počíta vl. vektor pre najväčšiu vl. hodnotu
- konvergencia je lineárna (t.j. exponent v $O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$ závisí lineárne od k)
- v závislosti od veľkosti podielu $\left|\frac{\lambda_2}{\lambda_1}\right|$ môžeme mať rýchlu konvergenciu ak je najväčšia vl. hodnota výrazne vyššia ako ostatné, v prípade, že majú podobnú veľkosť, dostaneme veľmi pomalú konvergenciu.

Našťastie, toto sa dá opraviť:

Inverzné iterovanie

Pre $\mu \in \mathbb{R}$, ktoré nie je vl. hodnotou matice A sú vl. vektory $(A-\mu I)^{-1}$ rovnaké ako vl. vektory A, zodpovedajúce vl. hodnoty $\{(\lambda_j-\mu)^{-1}\}$, kde $\{\lambda_j\}$ je spektrum A.

To vedie k nasledujúcej myšlienke:
Nech μ je blízko niektorej vl. hodnote λ_J. Potom $(\lambda_J-\mu)^{-1}$ môže byť výrazne vyššie ako $(\lambda_j-\mu)$ pre $j \neq J$, teda mocninové iterovanie pre $(A-\mu I)^{-1}$ môže konvergovať pomerne rýchlo ku q_J

Algoritmus 27.2 Inverzné iterovanie

$v^{(0)}$ = nejaký vektor $\|v^{(0)}\|=1$

pre $k=1,2,\dots$

riešiť $(A-\mu I)w = v^{(k-1)}$ pre w — vynásobiť $(A-\mu I)^{-1}$

$v^{(k)} = w/\|w\|$ — normalizovať

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleighov podiel

Tu prichádzajú viaceré otázky.

Čo ak je μ vl. hodnota A a $A-\mu I$ je singulárna?

Čo ak je μ skoro vl. hodnota, t.j. $A-\mu I$ je veľmi zle podmienená (t.j. riešenie $(A-\mu I)w = v^{(k-1)}$ bude zo zásady nepresné?)

$\rightarrow$ úloha 27.5. (DÚ)

- Aj inverzné iterovanie konverguje lineárne, ale vieme pomocou neho nájsť vl. vektory aj pre iné vl. hodnoty ako pre najväčšiu vl. hodnotu (to kontrolujeme vstupom μ).

Navyše, rýchlosť lineárnej konvergencie sa dá kontrolovať. — Závisí od "kvality" μ (a nie od štruktúry spektra $\{\lambda_j\}$).

Máme Tvrdenie:

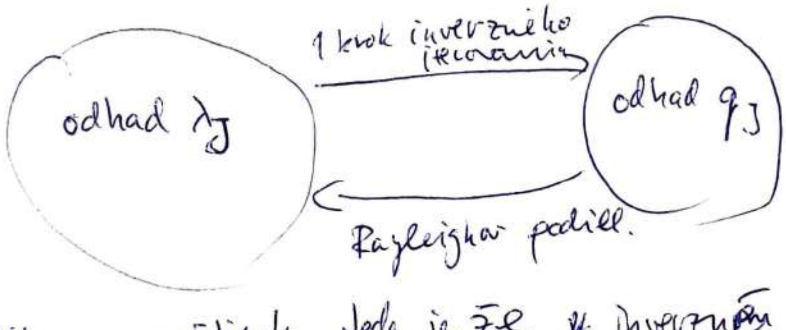
Nech λ_J je najbližšia vl. hodnota k μ a λ_K je druhá najbližšia, t.j. $|\mu-\lambda_J| < |\mu-\lambda_K| \le |\mu-\lambda_j|$ pre $j \neq J$.

Navyše nech $q_J^T v^{(0)} \neq 0$. Potom medzivýsledky algoritmu 27.2 spĺňajú
$$\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu-\lambda_J}{\mu-\lambda_K}\right|^k\right) \quad a \quad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu-\lambda_J}{\mu-\lambda_K}\right|^{2k}\right)$$
pre $k\to\infty$, $\pm$ má ~~podobný~~ rovnaký význam ako minule.

Ukazuje sa, že inverzné iterovanie je cenný nástroj numerickej lineárnej algebry, lebo je to štandardná metóda výpočtu vl. vektorov, ak sú vl. hodnoty (približne) známe. V tomto prípade sa náš algoritmus 27.2 používa tak, ako je napísaný, s výnimkou výpočtu Rayleighovho podielu.

Iterovanie pomocou Rayleighovho podielu (RQ)

Doteraz sme si ukázali metódu odhadu vl. hodnoty pre nejakého kandidáta na vl. vektor (Rayleighov podiel), a inú metódu ako z odhadnutej vl. hodnoty dostať vl. vektor (inverzné iterovanie).
Takže možnosť skombinovať tieto dve metódy sa prirodzene núka:



~~Tak~~ Hlavná myšlienka teda je, že v inverznom iterovaní vylepšíme tým, že v každom kroku vylepšíme odhad vl. hodnoty. Tento algoritmus sa nazýva <u>Rayleigh quotient algorithm</u> (algoritmus pomocou Rayleighovho podielu)

Algoritmus 27.3 (Rayleighov podiel)

$v^{(0)}$ = nejaký vektor s $\|v^{(0)}\|=1$
$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ = Rayleighov podiel pre $v^{(0)}$
pre $k=1,2,\dots$
  Riešiť $(A-\lambda^{(k-1)}I)w = v^{(k-1)}$ — vynásobiť $(A-\lambda^{(k-1)}I)^{-1}$
  $v^{(k)} = w/\|w\|$ — normalizovať
  $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — spočítať nový Rayleighov podiel

Konvergencia tohto algoritmu je výrazne lepšia ako inverzná iterácia – každá iterácia strojnásobí (!) počet presných cifier.

Veta ~~Algoritmus~~ Iterovanie pomocou Rayleighovho podielu konverguje k páru vl. vektor/vl. hodnota pre všetky počiatočné vektory $v^{(0)}$ s výnimkou množiny miery 0.
Ak konverguje, konvergencia je napokon kubická, t. j. ak λ_j je vl. hodnota a $v^{(0)}$ je dostatočne blízko k vl. vektoru q_j, tak

$$\| v^{(k+1)} - (\pm q_j) \| = O\left(\| v^{(k)} - (\pm q_j) \|^3 \right)$$

a

$$| \lambda^{(k+1)} - \lambda_j | = O\left(| \lambda^{(k)} - \lambda_j |^3 \right).$$

pre $k \to \infty$, znamienka $\pm$ nemusia byť rovnaké v

Dôkaz Dôkaz toho, že množina vektorov, pre ktorú alg. konverguje má kodimenziu ≥ 1 kniha neuvádza...
Ukážme aspoň konvergenciu, keď už k nej ~~p~~ dochádza:
Nech je (pre jednoduchosť) vl. hodnota λ_j jednoduchá (násobnosť $=1$)
Ak $\| v^{(k)} - q_j \| \leq \varepsilon$ pre dostatočne malé ε, potom Rayleighov podiel dáva odhad $\lambda^{(k)}$ s presnosťou $O(\varepsilon^2)$, t. j. $|\lambda^{(k)} - \lambda_j| = O(\varepsilon^2)$.
Potom pri nasledujúcom kroku inverznej iterácie $(v^{(k)}, \lambda^{(k)})$ máme

$$\| v^{(k+1)} - q_j \| = O\left(|\lambda^{(k)} - \lambda_j| \, \| v^{(k)} - q_j \| \right) = O(\varepsilon^3).$$

Navyše, konštanty vystupujúce v O budú rovnaké (rovnomerné) pre dostatočne malé okolia λ_j a q_j. takže dostaneme konvergenciu v štýle:

	$\| v^{(k)} - q_j \|$		$|\lambda^{(k)} - \lambda_j|$
k	ε	$\longrightarrow$	$O(\varepsilon^2)$
$k+1$	$O(\varepsilon^3)$	$\longrightarrow$	$O(\varepsilon^6)$
$k+2$	$O(\varepsilon^7)$	$\longleftarrow$	$O(\varepsilon^{18})$

z čoho máme tvrdenie vety.

Príklad Kubická konvergencia je taká rýchla, že si zaslúži príklad. | 2.5

Uvažujme maticu $A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix}$

a $v^{(0)} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

ak sa použije iterovanie pomocou Rayleighovho podielu pre A a $v^{(0)}$, vyjdú nasledujúce hodnoty $\lambda^{(k)}$: $\lambda^{(0)} = 5$

$\lambda^{(1)} = 5.2131\ldots$

$\lambda^{(2)} = 5.2143197431|184\ldots$

pričom presná hodnota pre vl. vektor najbližšie k $v^{(0)}$ je

$\lambda = 5.2143197433|77\ldots$ → presnosť na 10 číslic po 3 iteráciách

po ďalších troch iteráciách pomocou Rayleighovho podielu udávajú presnosť na 270 miest. — ak by sme to od našej mašiny chceli…

Počty operácií (len zbežne)

Uzavrime túto časť prednášky poznámkou o počte potrebných operácií:

Najprv nech A je všeobecná reálna matica typu $m \times m$.

- Každý krok mocninového iterovania si vyžaduje násobenie matica-vektor, t.j. $O(m^2)$ flopov.
- Každý krok inverzného iterovania si vyžaduje riešenie systému, čo na prvý pohľad dáva $O(m^3)$ flopov, ale dá sa to znížiť na $O(m^2)$ ak maticu predupravíme pomocou LU, QR alebo iného rozkladu.

Pri iterovaní pomocou Rayleighovho podielu sa však matica v každom kroku mení (odpočítavame od A iné $\lambda^{(k)} I$), teda

Spôsob ako vylepšiť/porazit $O(m^3)$ nie je až taký priamočiary

Tieto odhady sa však dramaticky zlepšia pre tridiagonálnu maticu:
Každá z iterácií si vyžaduje len $O(m)$ flopov v každom kroku.

Ak by sme pracovali s nesymetrickou maticou, musíme použiť Hessenbergov tvar z fázy 1 a tieto počty stúpnu na $O(m^2)$.

(Trefethen - Bau, lekcia 28)

QR-algoritmus bez posunov

QR algoritmus pochádza zo začiatku 60 tych rokov a v knihe ho označujú za jeden z "pokladov numerickej analýzy". V tejto lekcii ho prezentujeme v najjednoduchšej forme, ktorá sa dá chápať ako stabilný výpočet QR rozkladov matíc $A, A^2, A^3, \dots$

QR algoritmus

Základná verzia QR algoritmu vyzerá ~~absurdne~~ neuveriteľne jednoducho.

Algoritmus 28.1 "čistý" QR algoritmus

$A^{(0)} = A$

pre $k = 1, 2, \dots$

$Q^{(k)} R^{(k)} = A^{(k-1)}$ nájsť QR rozklad $A^{(k-1)}$

$A^{(k)} = R^{(k)} Q^{(k)}$ prehodiť poradie faktorov

Všetko, čo sa robí je urobiť QR rozklad, vynásobiť získané faktory Q, R v opačnom poradí RQ a opakovať.
Napriek tomu, za určitých podmienok, tento jednoduchý postup konverguje k Schurovmu tvaru pre všeobecné A, resp. diagonálnej matici, ak je A hermitovská.
Pre zjednodušenie veci, predpokladajme naďalej, že A je reálna symetrická, s reálnymi vl. hodnotami λ_j a ortonormálnymi vl. vektormi q_j.
Chceme teda ukázať, že matice $A^{(k)}$ budú konvergovať k diagonálnemu tvaru.

Aby konvergencia k diagonálnemu tvaru naozaj dala vl. hodnoty, vykonané operácie musia byť podobnostné transformácie. To, že $A^{(k-1)}$ a $A^{(k)}$ sú podobné sa ľahko overí:

$$Q^{(k)} R^{(k)} = A^{(k-1)} \quad \text{je to isté ako} \quad R^{(k)} = (Q^{(k)})^T A^{(k-1)}$$

Teda $$A^{(k)} = R^{(k)} Q^{(k)} = Q^{(k)T} A^{(k-1)} Q^{(k)}.$$

V skutočnosti, táto podobnostná transformácia sa už spomínala v lekcii 26, ako "zlá myšlienka", keď sme chceli A redukovať na horný trojuholníkový tvar priamo, v jednom kroku. Napriek tomu, ako krok iterácie sa ukáže vhodná.

V predošlej lekcii sme videli kubickú konvergenciu inverzného iterovania s Rayleighovho podielom, niečo podobné dosiahneme aj pre QR algoritmus, keď pozmeníme "čistý" QR algoritmus o posuny. Toto je jeden z troch krokov, ktoré myšlienku posúvajú k výpočtovej praxi:

(predpríprava) 1. Pred začiatkom iterovania, redukovať A na tridiagonálny tvar (lekcia 26)

(posun) 2. Namiesto $A^{(k)}$, faktorizovať posunutú maticu $A^{(k)} - \mu^{(k)} I$, kde $\mu^{(k)}$ je nejaký odhad vl. hodnoty

(deflácia) 3. Kedykoľvek to je možné (t. j. keď nájdeme vl. hodnotu), rozdeliť problém na diagonalizáciu podmatíc.

QR algoritmus s týmito vylepšeniami sa dá schematicky popísať:

Algoritmus 28.2 "Praktický" QR algoritmus

$(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ $\quad$ $A^{(0)}$ je tridiagonalizácia A

pre $k = 1, 2, \dots$

- vybrať posun $\mu^{(k)}$ $\quad$ napr. zvolme $\mu^{(k)} = A^{(k-1)}_{m,m}$ člen na diagonále
- $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} \underline{I}$ $\quad$ QR-rozklad $A^{(k-1)} - \mu^{(k)} \underline{I}$
- $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} \underline{I}$ $\quad$ Použiť faktory v opačnom poradí

Ak niektorý z mimo-diagonálnych zložiek $A^{(k)}_{j,j+1}$ je dostatočne blízka 0,

dať $A_{j,j+1} = A_{j+1,j} = 0$ a dostať

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^{(k)}$$

$\rightarrow$ aplikovať QR algoritmus na A_1 a A_2

– Takýto QR algoritmus s dobre zvolenými posunmi bol štandardnou metódou pre výpočet vl. hodnôt od začiatku 60-tych rokov. Až v 90-tych rokoch sa našiel zdatnejší súper – algoritmus "Rozdeľuj & Panuj" (Divide-and-conquer) (Lekcia 30)

- Tridiagonalizácia $\rightarrow$ lekcia 26
- posuny $\rightarrow$ lekcia 29
- deflácia $\rightarrow$ (bez ďalšieho komentára $\rightarrow$ CS)

Pozrime sa teda na "čistý" QR algoritmus a vysvetlime si, ako vlastne nájde vl. hodnoty.

Simultánne iterovanie bez normovania

Prístup pri skúmaní QR algoritmu bude taký, že sa pozrieme na inú metódu, o ktorej ukážeme, že je ekvivalentná. A tou je Simultánne iterovanie

Ak aplikujeme mocninové iterovanie na viacero vektorov (akoby na blok) dostaneme tzv. simultánne iterovanie, (resp. blokovú mocninovú metódu).

Predpokladajme, že máme n lin. nezávislých vektorov $v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$.

Ak $A^k v_1^{(0)}$ pre $k \to \infty$ konvergujú k vl. vektoru pre najv. vl. hodnotu (mocninové iterovanie), tak zovšeobecnením by bolo, že podpriestor $\langle A^k v_1^{(0)}, A^k v_2^{(0)}, \dots, A^k v_n^{(0)} \rangle$ by za istých predpokladov mohol konvergovať k podpriestoru $\langle q_1, q_2, \dots, q_n \rangle$ – generovanému n vl. vektormi $q_1, q_2, \dots, q_n$ zodpovedajúcimi n najväčším vl. hodnotám.

Maticovo:

$$V^{(0)} = \begin{bmatrix} | & & | \\ v_1^{(0)} & \cdots & v_n^{(0)} \\ | & & | \end{bmatrix}$$

potom

$$V^{(k)} = A^k V^{(0)} = \begin{bmatrix} | & & | \\ v_1^{(k)} & \cdots & v_n^{(k)} \\ | & & | \end{bmatrix}.$$

Keďže nás zaujíma stĺpcový priestor $V^{(k)}$, jeho dobrú bázu dostaneme pomocou (redukovaného) QR rozkladu $V^{(k)}$

$$\underset{m\times n}{\hat{Q}^{(k)}}\ \underset{n\times n}{\hat{R}^{(k)}} = \underset{m\times n}{V^{(k)}}$$

Teda by sme mohli očakávať, že za nie príliš silných predpokladov by sme ~~mali~~ dostali, že stĺpce

$\hat{Q}^{(k)}$ konvergujú k $\pm q_1, \pm q_2, \dots, \pm q_n$.

Toto očakávanie môžeme podoprieť analýzou podobnou analýze konvergencie mocninového iterovania.

po vyjadrení $v_j^{(0)}$ a $v_j^{(k)}$ pomocou vl. vektorov matice A:

$$v_j^{(0)} = a_{1j} q_1 + \dots + a_{mj} q_m$$

$$v_j^{(k)} = \lambda_1^k a_{1j} q_1 + \dots + \lambda_m^k a_{mj} q_m$$

ku konvergencii teda naozaj dôjde za predpokladu, že

1) $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \dots \geq |\lambda_m|$

(máme vl. hodnoty ostro usporiadané)

2) koeficienty a_{ij} sú "regulárne" v istom určitom zmysle

$\rightarrow$ zvoľme $\hat{Q} = \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \dots & q_n \\ | & | & & | \end{bmatrix}$

potom požadujme, aby v $\underset{n \times n}{\text{matici}}$ $\hat{Q}^T V^{(0)}$ boli hlavné minory regulárne, (t.j. ľavé horné podmatice $1\times1, 2\times2, \dots, n\times n$)

(Pozn - toto je podmienka existencie LU rozkladu)

<u>Tvrdenie</u> Predpokladajme, že pre iterovanie $V^{(k)} = A^k V^{(0)} = \hat{Q}^{(k)} \hat{R}^{(k)}$ sú splnené podmienky 1) a 2). Potom pre $k \to \infty$ stĺpce matic $\hat{Q}^{(k)}$ konvergujú lineárne k vl. vektorom A:

$$\| q_j^{(k)} - \pm q_j \| = O(C^k)$$

pre $1 \leq j \leq n$ a ~~[illegible]~~ $C < 1$ je konštanta $\max_{1 \leq k \leq n} |\lambda_{k+1}| / |\lambda_k|$.

(znamienko $\pm$ treba brať ako v minulej lekcii, odhad platí pre jeho správnu voľbu).

<u>Dôkaz</u>

Rozšírme $\hat{Q}$ ~~[illegible]~~ typu $m\times n$ na ortogonálnu maticu Q vl. vektorov, t.j. $A = Q \Lambda Q^T$. $\hat{Q}$ je zodpovedá prvým n stĺpcom Q, vezmime $\hat{\Lambda}$ ľavú hornú podmaticu typu $n\times n$ v matici Λ (teda hl. minor veľkosti $n\times n$) tiež diagonálna.

Potom:

$$V^{(k)} = A^k V^{(0)} = Q\Delta^k Q^T V^{(0)} = \hat{Q}\hat{\Delta}^k \hat{Q}^T V^{(0)} + \mathcal{O}(|\lambda_{n+1}|^k)$$

pre $k \to \infty$

Ak predpokladáme regularitu minorov $\hat{Q}^T V^{(0)}$, tak aj samotná $\hat{Q}^T V^{(0)}$ je regulárna, takže môžeme člen $\mathcal{O}(|\lambda_{n+1}|^k)$ vynásobiť sprava $(\hat{Q}^T V^{(0)})^{-1} \hat{Q}^T V^{(0)}$ a dostať

$$V^{(k)} = \left(\hat{Q}\Delta^k + \mathcal{O}(|\lambda_{n+1}|^k)\right) \hat{Q}^T V^{(0)}.$$

$\hat{Q}^T V^{(0)}$ je regulárna $\Rightarrow$ stĺpcový priestor $V^{(k)}$ je rovnaký ako stĺp. priestor $\hat{Q}\Delta^k + \mathcal{O}(|\lambda_{n+1}|^k)$.

Takže naozaj, stĺpcový priestor bude konvergovať lineárne k stĺpcovému priestoru matice $\hat{Q}$. (To by sa dalo kvantifikovať uhlami medzi podpriestormi - vzdialenosťou v $Gr(m,n)$)
(a rýchlosť zodpovedá $(|\lambda_{n+1}|/|\lambda_n|)^k$)

Predpokladali sme však, že nielen $\hat{Q}^T V^{(0)}$ je regulárna, ale aj jej všetky hlavné minory.

Zopakovaním vyššieuvedeného argumentu dostaneme podobnú konvergenciu aj pre lin. obaly prvých k stĺpcov $V^{(k)}$ k priestoru $\langle q_1, q_2, \dots, q_\ell \rangle$ pre $\ell = 1, 2, \dots, n$, z čoho vyplýva tvrdenie. $\boxtimes$

Simultánne iterovanie

Pre $k \to \infty$ každý z vektorov $v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)}$ konverguje k násobkom dominantného vl. vektora q_1 matice A, takže napriek tomu, že ich obal $\langle v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)} \rangle$ konverguje k niečomu užitočnému, takéto vektory tvoria veľmi zle-podmienenú bázu. *(ako)*

Ak by sme takéto iterovanie spravili vo floating-point aritmetike, užitočné "smery" by sa rýchlo stratili zaokrúhľovaním.

Záchrana je však jednoduchá, treba ortonormalizovať v každom kroku, nielen na konci. Takže nebudeme vytvárať $V^{(k)}$ podľa predpisu, ale budeme rátať s inými maticami $Z^{(k)}$ s rovnakým stĺpcovým priestorom.

Algoritmus 28.3 Simultánne iterovanie

zvoliť $\hat{Q}^{(0)} \in M_{m,n}(\mathbb{R})$ s ortonormálnymi stĺpcami

pre $k = 1, 2, \ldots$

$$Z = A\hat{Q}^{(k-1)}$$

$$\hat{Q}^{(k)}\hat{R}^{(k)} = Z \qquad \text{redukovaný QR rozklad matice } Z$$

Zo zápisu vidíme, že stĺpcové priestory $\hat{Q}^{(k)}$ a Z sú rovnaké a obe sa rovnajú stĺpcovému priestoru $A^k\hat{Q}^{(0)}$.

Z matematického hľadiska teda tento nový predpis ~~konverguje~~ pre simultánne iterovanie konverguje rovnako, ako predošlý.

<u>Tvrdenie</u> Algoritmus 28.3 generuje rovnaké matice $\hat{Q}^{(k)}$ ako iterácia

$V^{(k)} = A^k V^{(0)} = \hat{Q}^{(k)}\hat{R}^{(k)}$ (za predpokladu $\hat{Q}^{(0)}$ sú rovnaké).

Pri predpoklade usporiadania vl. hodnôt a regularity minorov máme aj konvergenciu z predošlého tvrdenia.

Simultánne Iterovanie $\Leftrightarrow$ QR algoritmus

Teraz sa dostávame k vysvetleniu podstaty QR algoritmu.

Ide o simultánne iterovanie s plnou sadou m vektorov, menovite, začne sa s $Q^{(0)} = \underline{I}$.

Keďže matice $\hat{Q}^{(k)}$ sú teraz štvorcové, v každom kroku máme úplné QR rozklady a v súlade so značením zo začiatku knihy (Lekcia 7) môžeme dať preč "striešku".

Mali by sme ale rozlíšiť matice $\hat{Q}^{(k)}$ zo simultánneho iterovania ($\underline{Q}^{(k)}$) a $Q^{(k)}$ z QR algoritmu.

Porovnajme teda tieto dva postupy a pridajme ešte štvrtý vzorec pre matice $A^{(k)}$, resp. $\underline{Q}^{(k)}$.

Simultánne iterovanie:

$$(*)\quad \left.\begin{aligned} \underline{Q}^{(0)} &= I \\ Z &= A\,\underline{Q}^{(k-1)} \\ Z &= \underline{Q}^{(k)} R^{(k)} \end{aligned}\right\}$$

$$A^{(k)} = (\underline{Q}^{(k)})^T A\, \underline{Q}^{(k)}$$

QR algoritmus bez posunov

$$(**)\quad \left.\begin{aligned} A^{(0)} &= A \\ A^{(k-1)} &= Q^{(k)} R^{(k)} \\ A^{(k)} &= R^{(k)} Q^{(k)} \end{aligned}\right\}$$

$$\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \dots Q^{(k)}$$

Pre oba algoritmy definujme ešte $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \dots R^{(1)}$

O ekvivalencii týchto dvoch algoritmov vypovedá tvrdenie:

<u>Tvrdenie</u> Procesy $(*)$, $(**)$ generujú rovnaké postupnosti matíc $\underline{R}^{(k)}, \underline{Q}^{(k)}$ a $A^{(k)}$, ktoré ~~sú stĺpce~~ zodpovedajú QR rozkladu mocniny matice A: $A^{k} = \underline{Q}^{(k)} \underline{R}^{(k)}$

Spolu s ~~projekciou~~ podobnosťou

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

Dôkaz postupujme indukciou vzhľadom na k.

Pre $k=0$ máme:

$$A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I \quad a \quad A^{(0)} = A.$$

Pre $k \geq 1$ sa pozrime na simultánne iterovanie (*)

$$A^k = A A^{k-1} \overset{IP}{=} A \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

Pre $k \geq 1$ pre QR algoritmus (**) máme:

$$A^k = A \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} Q^{(k)} R^{(k)} \underline{R}^{(k-1)}$$

$\underline{Q}^{(k-1)} A^{(k-1)}$

$$= \underline{Q}^{(k)} \underline{R}^{(k)}$$

a tiež

$$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)} = (Q^{(k)})^T (\underline{Q}^{(k-1)})^T A \underline{Q}^{(k-1)} Q^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

$\boxed{EB}$

Konvergencia QR algoritmu

Teraz máme všetko potrebné k tomu, aby sme mohli povedať pomerne veľa o konvergencii QR algoritmu.

Predchádzajúca veta dáva kvalitatívne odpovede:

- prvá rovnica hovorí, že konštruujeme ortonormálne bázy stĺpcového priestoru matíc A^k → teda by sme mali nájsť vl. vektory.
- druhá rovnica hovorí, ~~[illegible]~~ prečo by sme mali nachádzať vl. hodnoty.

diagonálne zložky $A^{(k)}$ sú Rayleighove podiely pre stĺpce matice $\underline{Q}^{(k)}$. Keďže tieto stĺpce konvergujú k vl. vektorom, diagonála (Rayleighove podiely) bude konvergovať (dva krát tak rýchlo) k vl. hodnotám.

Tiež, mimodiagonálne členy zodpovedajú tzv. zovšeobecnenému Rayleighovmu podielu pre rôzne vl. vektory zľava a sprava $(q_i^{(k)})^T A q_j^{(k)}$ by mali ísť k nule.
Z ortogonality vl. vektorov, tieto

Rovnice $A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$, $A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$
sú ľahko zapamätateľné a všetko podstatné o QR-algoritme z nich vyplýva.

Pre kvantitatívne pochopenie konvergencie máme nasledujúci dôsledok tvrdenia o konvergencii simultánneho iterovania:

<u>Tvrdenie</u> Použime čistý QR algoritmus na reálnu symetrickú maticu, pre ktorú $|\lambda_1| > |\lambda_2| > \dots > |\lambda_m|$, a ktorej matica vlastných vektorov Q má všetky hlavné minory regulárne. Potom pre $k \to \infty$ $A^{(k)}$ konverguje lineárne s konštantou $\max_k |\lambda_{k+1}| / |\lambda_k|$ k diagonálnej matici $\mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ a $\underline{Q}^{(k)}$ (s patrične prispôsobenými znamienkami) konverguje ku Q rovnakou rýchlosťou.