Linjär algebra och numerisk analys, 2019

Bonusuppgift nummer 2: Wavelet-analys för datakomprimering, maximalt 4 bonuspoäng

Wavlet-analys är en teknik för approximation av data som används i många sammanhang t.ex. inom bild- och signalbehandling och vid numerisk approximation av differentialekvationer. Exempelvis använder JPEG wavelets för bildkomprimering i sitt JPEG2000-schema. Kursen TMA462 Wavelet analysis behandlar waveletanalys grundligt.

Här gör vi en mycket enkel elementär introduktion av wavelet-tekniken för komprimering av data. Idén bygger på upprepade medelvärdesberäkningar med hjälp av matrismultiplikation.

Vi inleder med ett exempel. Låt $v = [v_1\ v_2\ v_3\ v_4]^T$ vara en datavektor och låt $A = [\frac{1}{2}\ \ \frac{1}{2}]$, $B = [\frac{1}{2}\ \ -\frac{1}{2}]$ och $Z = [0\ \ 0]$ vara 1×2 -matriser. Med 4×4 -matrisen $A_1 = \begin{bmatrix} A & Z \\ Z & A \\ B & Z \\ Z & B \end{bmatrix}$ får vi $u = A_1 v = \begin{bmatrix} (v_1+v_2)/2 \\ (v_3+v_4)/2 \\ v_1-(v_1+v_2)/2 \\ v_3-(v_3+v_4)/2 \end{bmatrix}$. De två första elementen i vektorn u är alltså *medelvärden* och de två sista är avstånd till medelvärden, här kallade *distansvärden.* För fortsatt medelvärdesberäkning mellan de två första elementen i u använder vi matrisen $A_2 = \begin{bmatrix} A & Z \\ B & Z \\ Z & \\ Z & I \end{bmatrix}$, där I är enhetmatrisen av ordning 2×2.

Nu blir $w = A_2 A_1 v = A_2 u = \begin{bmatrix} (u_1+u_2)/2 \\ u_1-(u_1+u_2)/2 \\ u_3 \\ u_4 \end{bmatrix}$, där w_1 är medelvärde och w_2 är distansvärde. För en vektor v av längd $n = 2^p$, p ett positivt heltal, kan idén i exemplet generaliseras till upprepade medelvärdesberäkningar på p nivåer.

I nästa exempel förklaras hur datakomprimeringen utförs. Betrakta följande data då $n = 2^3 = 8$: $v = [37\ 33\ 6\ 16\ 26\ 28\ 18\ 4]^T$. Med $A_1 = \begin{bmatrix} A & Z & Z & Z \\ Z & A & Z & Z \\ Z & Z & A & Z \\ Z & Z & Z & A \\ B & Z & Z & Z \\ Z & B & Z & Z \\ Z & Z & B & Z \\ Z & Z & Z & B \end{bmatrix}$ får vi

$A_1 v = [35 \;\; 11 \;\; 27 \;\; 11 \;\; 2 \;\; -5 \;\; -1 \;\; 7]^T$. Nästa steg blir med $A_2 = \begin{bmatrix} A & Z & Z & Z \\ Z & A & Z & Z \\ B & Z & Z & Z \\ Z & B & Z & Z \\ Z & Z & & \\ Z & Z & I & \\ Z & Z & & \\ Z & Z & & \end{bmatrix}$

och $A_2(A_1 v) = [23 \;\; 19 \;\; 12 \;\; 8 \;\; 2 \;\; -5 \;\; -1 \;\; 7]^T$. Sista rekursiva medelvärdesberäkningen blir

nu med $A_3 = \begin{bmatrix} A & Z & Z & Z \\ B & Z & Z & Z \\ Z & & & \\ Z & & & \\ Z & & & \\ Z & & I & \\ Z & & & \\ Z & & & \end{bmatrix}$ och $w = A_3(A_2(A_1 v)) = [21 \;\; 2 \;\; 12 \;\; 8 \;\; 2 \;\; -5 \;\; -1 \;\; 7]^T$.

All information för att återskapa v finns i w och matriserna A_1, A_2 och A_3. Man kan ju formellt beräkna $v = A_1^{-1}(A_2^{-1}(A_3^{-1}w))$.

Datakomprimeringen bygger nu på att små **distansvärden**, element till belopp mindre än eller lika med ett tröskelvärde ϵ, i vektorn w ersätts med nollor, som inte behöver lagras. **OBS!!** Detta gäller alltså bara distansvärden inte medelvärden. Med $\epsilon = 3$ i vårt exempel får vi $\tilde{w} = [21 \;\; 0 \;\; 12 \;\; 8 \;\; 0 \;\; -5 \;\; 0 \;\; 7]^T$ och $\tilde{v} = A_1^{-1}(A_2^{-1}(A_3^{-1}\tilde{w})) = [33 \;\; 33 \;\; 4 \;\; 14 \;\; 29 \;\; 29 \;\; 20 \;\; 6]^T$.

Uppgifter:
Gör en wavelet-approximation av diskreta värden för funktionen $f(x) = e^x \cos(\pi x)$ på intervallet $[0\ ,\ 3]$. Välj n ekvidistanta funktionsvärden inklusive ändpunkterna av intervallet. Undersök de tre fallen:
1) $n = 16,\ \epsilon = 1.5$
2) $n = 32,\ \epsilon = 1.0$
3) $n = 64,\ \epsilon = 0.5$
a. Skriv en MATLAB-funktion för wavelet-approximationen. I stället för invertering av matriser löser du förstås linjära ekvationssystem, *backslash* i MATLAB.
b. Rita ut funktionen $f(x)$ och approximationen $\tilde{v}$ i de tre fallen.
c. Ange felet på formen $\frac{1}{n}\|v - \tilde{v}\|_2$ i de tre fallen.
d. Ange komprimeringsgraden dvs. antalet icke-nollor i $\tilde{w}$ dividerat med antalet icke-nollor i w för de tre fallen.

Inlämning. Funktionsfilen, grafer, felnormer enligt c)-uppgiften och komprimeringsgrader.